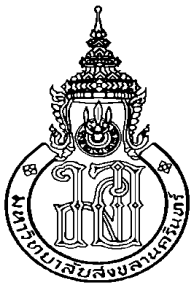


Sentiment Analysis of the Burmese Language using the Distributed Representation of *n*-gram-based Words

Myat Lay Phyu

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

Master of Science in Information Technology

Prince of Songkla University

2018

Sentiment Analysis of the Burmese Language using the Distributed Representation of *n*-gram-based Words

Myat Lay Phyu

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Information Technology
Prince of Songkla University
2018

Thesis Title	Sentiment Analysis of the Burmese Language using the Distributed Representation of *n*-gram-based Words
Author	Miss Myat Lay Phyu
Major Program	Information Technology

Major Advisor

……
(Prof. Dr. Kiyota Hashimoto)

Examining Committee:

…….Chairperson
(Dr. Nattapong Tongtep)

……
(Prof. Dr. Kiyota Hashimoto)

……
(Dr. Thatsanee Charoenporn)

The Graduate School, Prince of Songkla University, has approved this thesis as partial fulfillment of the requirements for the Master of Science Degree in Information Technology.

……
(Prof. Dr. Damrongsak Faroongsarng)
Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

_____ Signature

(Prof. Dr. Kiyota Hashimoto)
Major Advisor

_____ Signature

(Ms. Myat Lay Phyu)
Candidate

I hereby certify that this work has not been accepted in substance for any degree and is not being currently submitted in candidature for any degree.

_____ Signature

(Ms. Myat Lay Phyu)
Candidate

Thesis Title Sentiment Analysis of the Burmese Language using the Distributed Representation of *n*-gram-based Words
Author Miss Myat Lay Phyu
Major Program Information Technology
Academic Year 2017

ABSTRACT

Due to the availability of people's opinions and customer reviews, the need to analyze those texts have been more important. Sentiment analysis, or opinion mining, estimates their polarity, whether they are positive or negative, using machine learning techniques. Many methods have been proposed but they assume the basic preprocessing of text data including word segmentation and word sentiment values. However, such preprocessing is not easily available for low resource languages such as Burmese, Khmer and Lao due to the unavailability of annotated big corpora and basic natural language processing tools. The objective of this research is to solve these difficulties of low resource language processing. The goal is to propose an effective and efficient method to enable sentiment analysis without considering language specific characteristics. The scope of the research is the languages without word boundaries in written text, specifically Burmese. The methodology consists of two proposals, a character-based variable-length n-gram word model and a word grouping method with word similarities calculated with distributive word representation models. The proposed method is compared with Conditional Random Field (CRF) baseline approach, which is also proposed newly in this thesis, and achieved a similar result as the CRF-based word segmentation with a small size of supervised data. The proposed method is also validated with a larger size of data using Amazon product reviews. Thus, the proposed methods in this thesis provide an effective and efficient way for low resource language processing without focusing on language specific characteristics.

Keywords—Sentiment analysis, Burmese language, variable-length n-gram word model, CRF, distributive word representation model

ACKNOWLEDGEMENT

I would like to say thank you to all people who contributed in some way in this thesis work. Firstly, I would like to thank Higher Education Research Promotion and the Thailand's Education Hub for Southern Region of ASEAN Countries Project Office of the Higher Education Commission for the scholarship. In addition, I would like to present my special thanks to my thesis advisor, Prof. Dr. Kiyota Hashimoto. Not only he gives advices for my thesis, but also, he supports technical devices to do my research. I can consult him whenever I have troubles with my research and other problems. I learn a lot about my research field and other knowledge from my advisor.

Secondly, besides my advisor, I would like to express my grateful thanks to all my thesis committee members: Dr. Nattapong Tongtep and Dr. Thatsanee Charoenporn for their excellent comments, and suggestions. I also would like to thank to all lecturers for teaching other related courses in Master study and staffs of Master of Science in Information Technology, College of Computing for their kindly help. Moreover, I would like to thank to all the participants when I made questionnaire surveys in Myanmar for their kind cooperation and taking the time to assist me in my educational endeavour.

Finally, I would like to present my gratitude to my beloved family and friends for their wonderful encouragement and support throughout the Master study. For all those whose names are not listed here, I would like to express my warmest thanks for their valuable assistance.

Myat Lay Phyu

Contents

Contents (cont.)

Contents (cont.)

Contents (cont.)

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background and Introduction

Nowadays, an increasing number of people are eager to use the Internet. They are using because of many reasons: they may want to obtain the knowledge or news around the world; they may want to share or express their feeling or opinion on the social media and communicate with many people around the world, etc. Since today is the age of technology, people can know a vast amount of information online just by clicking on their smartphone or other devices.

Indeed, opinions of others have a significant influence on our decision-making process. People consider opinions of other people to make some decisions in daily choices of products and services, tourism destinations, business, and political standpoints. Before the age of the Internet, people asked their friends or relatives about them or consulted newspapers, magazines, and book. However, it is much easier now to get different opinions from different people around the world via different sources such as websites or social media. It is desirable, but on the other hand, the amount of information online has already reached the level that we cannot read all the relevant writings. Thus, various techniques and methods of extracting and summarizing important parts of the information have been investigated. Sentiment analysis, or opinion mining, is a research topic of natural language processing and related fields to estimate the polarity of text, whether a passage or a sentence is positive or negative as a whole or according to a specific viewpoint (Feldman, 2007).

In general, language processing research area has been popular, and many researchers have proposed a variety of Natural Language Processing (NLP) tasks and their solutions. Typical basic NLP tasks include word segmentation, named entity recognition, part-of-speech tagging, shallow or deep syntactic parsing, to name a few. Word segmentation means segmenting the text into a sequence of words. Named entity recognition identifies and labels the sequence of words in a text which are normally the names of people, organizations or locations. Part-of-speech tagging gives the part of speech of each word in texts. Syntactic parsing estimates the grammatical structure of a sentence. All these tasks are usually essential to more purpose-oriented natural

language processing tasks like sentiment analysis, automatic summarization, machine translation. In the past, these were tackled with rule-based systems in which all the rules were set by humans according to the human experience and wisdom. Since around 1980s, together with the easier availability of bigger data, machine learning has been extensively employed to improve the accuracy of these tasks.

However, a considerable size of high-quality data is a must for most machine learning studies for these basic language processing tasks. High quality means that the data must be well-designed, checked carefully, and given the correct answers with which machine learning tries to learn from the data. For English, Chinese, Japanese, and other so-called major languages, many corpora, language datasets, are already available, and basic language processing tools based on them are also easily available. This is not the case with many minor languages, including many South-East Asian languages like Burmese, Khmer, Lao, Thai, to name a few. These languages are often called low resource languages. Lack of resources on both data and basic tools makes it a hard challenge to pursue various language processing tasks satisfactorily.

To tackle with this issue of lack of resources in many languages, there are roughly two approaches. One is to prepare necessary data and develop basic tools. Although it is becoming possible now to rely more on machine learning with gigantic sizes of data, this approach, though necessary, costs much of human efforts and financial investments. The other is to develop workaround methods without using such resources. The latter may inherently be not able to achieve as good results as the first approach, but it is worth pursuing for obviously realistic reasons. This thesis pursues the latter, particularly for the Burmese language as an example of low resource languages.

Let us consider the process of sentiment analysis. First, we need a considerable size of text data. Data may come from officially written texts or from social media texts. Regardless of the sources, the data to be processed must contain the sentiment polarity, whether they are positive, neutral, or negative, They may be assigned manually or the corresponding rating by the writer or readers may be employed. Manual assignment costs a lot, while corresponding ratings, as well as text data, may not be of sufficient quantity for machine learning in the case of low resource languages. The text data need a variety of preprocessing, including noise reduction, normalization, sentence segmentation, word segmentation, lemmatization, part-of-speech tagging, etc. As mentioned, most of them are not easily available for low resource languages. Burmese, Khmer, Lao, and Thai, in particular, have difficulty in word segmentation and part-of-speech tagging due to their linguistic characteristics. Because of the growing market in developing countries and the emergence of diverse interests in products, services, business, and politics, there is a growing

demand for sentiment analysis for these low resource languages now. Thus, if it takes more time and money to prepare good resources, and unfortunately it is the case in many low resource languages, some workaround techniques are in urgent demands.

In this thesis, I pursue research for the development of word segmentation for sentiment analysis without employing ordinary word segmentation tools. The main target is the Burmese language. First, Burmese is written without explicit spaces for segmenting the words correctly, same with Khmer, Lao, and Thai. Second, there are few publicly available word segmentation tools (Natural Language Processing Lab, 2011). Third, high-quality datasets for sentiment analysis are not available. Fourth, there are also few publicly available sentiment dictionaries, which are necessary for most sentiment analyses. Chen and Steven (2014) proposed to construct high quality sentiment lexicons for 136 languages. Among them, the Burmese language lexicon contains 461 words. This is the reality not only for Burmese but also for many low resource languages. To tackle this issue, I construct a small size of Burmese newspaper article datasets with questionnaire surveys to assign sentiment values to each article. I also employ a larger size of English sentiment dataset to construct a pseudo-Burmese dataset for evaluation. For word segmentation, I construct a CRF-based word segmentation classifier as a baseline method with a tiny size of manually tagged dataset, which is also an achievement in this thesis. The main proposal of this thesis is the employment of a character-based variable-length n-gram word segmentation with word grouping based on distributive word representation models. N-gram word segmentation has often been used for text analysis, particularly when word segmentation tools are not available or when the target dataset contains too many spelling variations and new or instantaneous expressions as in many social media texts. However, simple n-gram word segmentation produces a much larger number of words compared to ordinary word segmentation, which raises issues of sparsity. I employ a variable-length method with which different sizes of n-gram words are automatically produced, which reduces the number of n-gram words. To further reduce the number of n-gram words, I propose employment of distributive word representation models to calculate the similarity among n-gram words, in particular, word2vec (Mikolov, et al., 2013; Mikolov, et al., 2013) and GloVe Pennington, Socher and Manning (2014). For the sentiment dictionary construction, SO-LSA (Semantic Orientation-Latent Semantic Analysis) (Turney and Littman, 2003) method is employed to calculate the sentiment value of words in the dataset. The employment of distributive word representation models and the SO-LSA method are first investigated in this thesis. All the proposals above are then evaluated with experiments using Support Vector Machines (SVM) (Weston, 2006) for sentiment analysis classification. The result shows that pseudo-word

approach achieved similar result with an ordinary word approach, which enables to investigate low resource languages without developing language-specific tools and techniques.

The organization of this thesis is as follows. Chapter 2 surveys relevant literature on sentiment analysis and word segmentation. Chapter 3 proposes my methodology. Chapter 4 evaluates a CRF-based word segmentation. Chapter 5 and 6 evaluate sentiment analysis based on my proposals with different conditions compared. Chapter 7 is the conclusion.

In this study, the Burmese language, one of the low resource languages, is used as the target language. Since I am a Burmese, the Burmese language processing is in my best interest, as well as easiness to check the methodology. However, all of my proposed techniques are not just dedicated to the Burmese language but can be applied to any low resource languages, particularly those without explicit word boundaries. Though it is a future task to investigate the effect of my proposed methods in different low resource languages, the contribution of this thesis is well expected to be extended to other tasks in the Burmese language as well as to other low resource languages.

1.2 Objective

1) To develop and evaluate a method to make a sentiment analysis of the Burmese language.
2) To devise techniques of deep learning for low resource languages.

1.3 Scope

This research focuses on sentiment analysis of newspaper articles written in the Burmese language, which are extracted from online. The proposed methods are designed to cope with the difficulties of text processing of low resource languages.

1.4 Expected Outcome

A word segmentation method without consulting tagged corpora or existing tools is established for the Burmese language. This method does not consult linguistics characteristics of Burmese, and so it is highly expected that the same method is applicable to any low resource languages.

The proposed word segmentation method, together with a word similarity calculation method using distributive word representation models achieves the result similar to ordinary word segmentation for a sentiment analysis task, which is expected to be applied to various data in low resource language successfully.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Generally, the main steps in sentiment analysis processing are data segmentation, calculation of word sentiment values, sentiment dictionary creation, and binary classification or summative sentiment calculation. In this chapter, I am going to explain the nature of the Burmese language, the necessity of word segmentation and different approaches of word segmentation, a machine learning framework, distributive word representations models and the process of sentiment classification.

2.2 The Burmese Language

2.2.1 Nature of Burmese Language

The Burmese language is the official language of the Republic of the Union of Myanmar (henceforth, Myanmar). Myanmar, also known as Burma, is a sovereign state located in Southeast Asia and is bordered by China, Thailand, India, Laos, and Bangladesh. Burma has a population of nearly 52 million, and there are eight main races such as Burma, Kachin, Kayar, Kayin, Chin, Mon, Yakhine, and Shan. The Burmese language is spoken by two-thirds of the population, approximately 32 million, as the first language, and by 10 million people as a second language. Other languages are also spoken in Myanmar such as Kachin, Kayin, Chin, Mon, Yakhine, and Shan.

The Burmese language has 33 main alphabets or consonants (some people say 34 because one consonant has two types), 7 independent vowels, 7 dependent vowels, 4 medials, 2 final symbols, 2 tone marks, 4 abbreviations, 2 types of punctuation and numerals shown in Figure 2.1 (The Library of Congress, 2011). The Burmese text is written from left to right. It requires no explicit space between words, although modern writing usually contains spaces after each clause or meaningful unit to enhance readability. However, there is no specific rule for space adding. Consonants are combined with vowels, medials, final symbols and tone marks to form a complete

meaning of words. Vowels and medials are attached front, back, above and below of the consonant. Final symbols are attached above the consonant. Tone marks are always attached below the consonant and at the end of one syllable, that is, after the consonant and all the attached back vowels. Vowels can be grouped into two categories: dependent vowels and independent vowels. Dependent vowels are combined with consonants to form a word. Most independent vowels are standalone, and some are combined with other characters. The symbols to be attached with a consonant is almost mandatory to form a word, and only a few consonants can be standalone.

Burmese belongs to the Sino-Tibetan language family. The Burmese lexicon had a deep and lasting influence from the classical Indian languages, namely Sanskrit, associated with Hinduism and Mahayana Buddhism, and Pali, associated with Theravada Buddhism. There are two ways for loanwords in Burmese from Pali, either directly from Pali texts or Mon. There are also many English loanwords. The influence of English is particularly typical with technical terms. Most of the words of technological in modern Burmese is directly borrowed from English (Jenny and Tun, 2016).

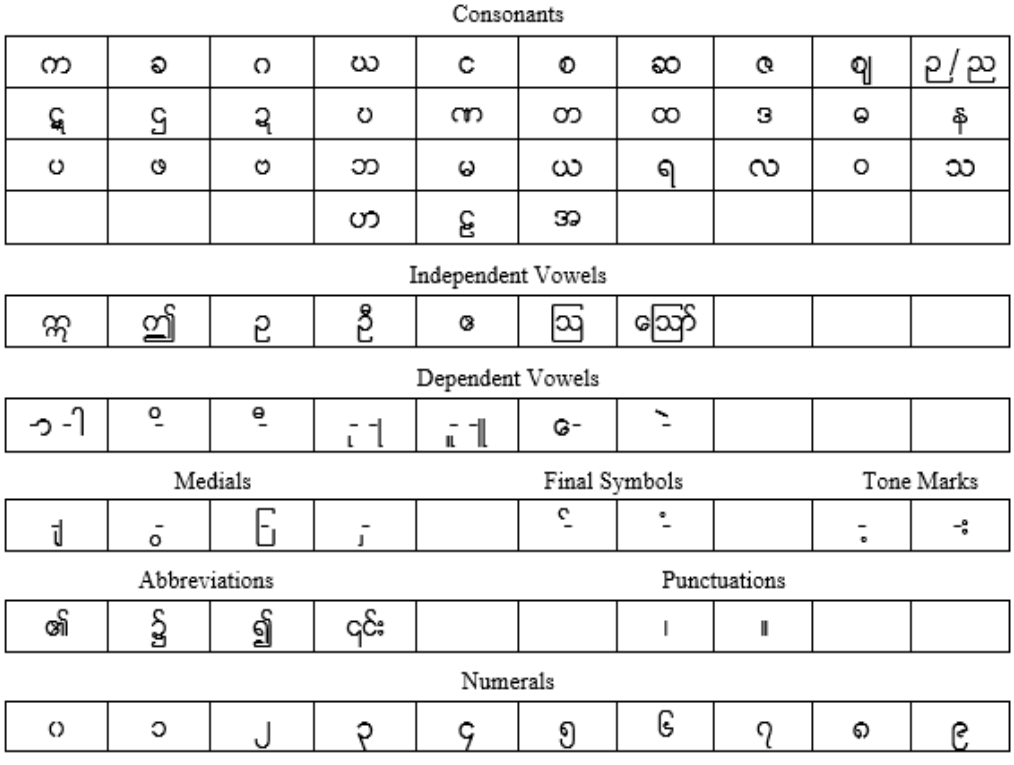


Figure 2.1 Burmese Characters

2.2.2 Myanmar and Burma: Formal and Colloquial Burmese

The Burmese language has two distinct stylistic varieties, colloquial and literary. The first one used in ordinary conversation, while the second is used officially in writing as well as speech in formal contexts, such as public speeches and state media broadcasts. In recent years, people have written in the colloquial style in the domains such as novels, journals, and newspaper articles which are previously covered by the literary style and media broadcasts, especially in non-government and foreign channels such as the BBC and VOA. In some types of text, such as novels, magazine articles, and advertisements, the text is frequently a mixture of both. The main difference between literary and colloquial styles is that the use of forms of grammatical function words and to some extent lexicon. The literary style is written by using more consistent grammatical function words such as subject and object markers than the colloquial style. The competing names of the country, namely Burma vs. Myanmar, represent the two styles. The form *myanma* or *myama* belongs originally to the literary style, while *bama* is in the colloquial style. The English name Burma came from *bama*. In 1989, the Burmese government changed the English name of the country from Burma to Myanmar, coming from *myanma* or *myama*. The word 'Myanmar' is used to refer to the country, while 'Burmese' is used for the people and the language (Jenny and Tun, 2016).

2.2.3 Word Order

The basic word order in the Burmese language is Subject-Object-Verb, which is different from English or Thai. For example, 'I like apple' in English can be said to 'I apple like' in Burmese. The verb and its modifiers appear at the final position in a sentence, while all the other elements are ordered rather freely before it. Likewise, Burmese has postpositions, instead of prepositions. In Burmese, 'Table beside' means 'Beside the table'. Noun modifiers and relative clauses come before the modified noun. For example, the English sentence 'the books which I gave you' corresponds to 'I you gave which books' in Burmese. The clause linkers, or subordinators, comes at the end of the clause. This is also true of any type of clause. Thus 'I did not come because I did not have time' in Burmese is 'I time not have because I not come' (Jenny and Tun, 2016).

2.2.4 Script and Numerals

In the Burmese language, there is no distinction between the uppercase and lowercase letters in the same way as Thai. Punctuation is restricted to two marks which correspond

to a comma and a full stop in the English language. Although Western numerals are more and more used in the written Burmese language, the Burmese numerals are still commonly used in everyday life. The basic system and arrangement of the Burmese numerals are the same with English numerals, only the shapes of the numerals are different (Jenny and Tun, 2016).

2.2.5 Markers and Particles

A great number of markers and particles are commonly used in the Burmese text. There is a distinction between markers and particles based on their operations on the phrase level and those on the clause or sentence level. However, this distinction is not absolute, and there are some markers and particles that are difficult to classify.

Phrasal markers are attached to noun phrases or verb phrases to express a wide range of grammatical functions.

Clausal markers can be used to express the relations between clauses or specify the grammatical function of a clause or sentence. They are similar to English subordinators such as ‘that’, ‘because’, ‘though’, ‘if’ and ‘when’. However, the position is different with English subordinators: the clausal markers always appear at the end of the subordinate expression.

Plural markers, which are suffixed to nouns and pronouns, express different functions and have different syntactic distribution.

Pragmatic particles are used to express a wide range of pragmatic discourse functions. They appear on the phrase level as well as on the clause and sentence level, and the same particle may be either phrasal or clausal in many cases. The difference with grammatical markers is that they do not express fixed grammatical notions and are generally optional.

Phrasal particles add pragmatic discourse information to noun phrases, verb phrases, or adverbials and express notions such as contrast with other referents or situations.

Clausal particles usually occur at the end of a clause or sentence which express different notions such as insistence and general emphasis (Jenny and Tun, 2016).

2.2.6 Overview of Burmese Natural Language Processing

Natural Language Processing of the Burmese language has been studied from the 1990s. The authors have proposed different kinds of NLP tasks. There are not many works for the Burmese language processing and the Burmese language is one of the low resource languages.

For corpus construction, Okell (2018) compiled a corpus of modern Burmese in the 1990s, and it was converted into the Unicode format in 2018. The corpus size is 8.8.MB. Htike,

Pa, and Thu (2017) published Myanmar Part-of-Speech Corpus. The corpus consists of 11,000 sentences (264,920 words or 242,865 words if some words are grouped as compound words) which are manually word segmented and POS tagged. Myanmar corpora are not widely publicly available in Myanmar. The size of the available corpus is not large, and they are not sufficient to perform natural language processing tasks. Some of the proposed approaches for different kinds of Burmese language processing are as follows.

For part-of-speech tagging, Myint, et al. (2011) analyzed the syntactic structure of Myanmar grammatical categories to be able to tag the words in Myanmar text with standard Part-of-Speech (POS) tags. They developed 27 customized lexical rules in order to propose finer or standard POS tags from basic POS tags combinations by analyzing Myanmar grammatical categories. Myint, et al. (2011) proposed a method that segments the input sentence into meaningful words and gives appropriate POS tags to these words. Zin and Thein, (2009) proposed a machine learning algorithm for Myanmar Tagging using a corpus-based approach. Their method was a combination of supervised and unsupervised learning which use pre-tagged and untagged corpus respectively. The proposed method of Myint, (2011) was to segment the input sentence into words by using segmentation rules and these words are assigned with appropriate syntactic categories of Myanmar language by using rule-based and probabilistic approach. These papers are listed in Table 2.1.

The process of sentiment analysis was performed in Aung (2016); Aye and Aung (2017); Thant, et al. (2017). Aung (2016) processed sentiment analysis by collaboration with opinion extraction, summarization, and tracking the records of teachers. They analyzed the text comments written by students using lexicon-based sentiment analysis to predict teacher performance. Aye and Aung (2017) proposed to create Myanmar sentiment lexicon for food and restaurant domain and analyzed the customers' reviews by using lexicon-based sentiment analysis for the recommendation. Thant, et al. (2017) developed a system to assign polarity scores to Facebook Myanmar movie comments. These papers are listed in Table 2.2.

Table 2.1 POS Tagging

Authors	Title of the paper/project	Year
Zin and Thein (Zin and Thein, 2009)	Part of Speech Tagging for Myanmar Using Hidden Markov Model	2009
Myint (Myint, 2011)	A Hybrid Approach for Part-Of-Speech Tagging of Burmese Texts	2011
Myint, Htwe and Thein (Myint, et al., 2011)	Bigram Part-of-Speech Tagger for Myanmar Language	2011
Myint, Htwe and Thein (Myint, et al., 2011)	Normalization of Myanmar Grammatical Categories for Part-of-Speech Tagging	2011

Table 2.2 Sentiment Analysis

Authors	Title of the paper/project	Year
Aung (Aung, 2016)	A Lexicon Based Sentiment Analyzer Framework for Student-Teacher Textual Comments	2016
Aye and Aung (Aye and Aung, 2017)	Sentiment Analysis for Reviews of Restaurants in Myanmar Text	2017
Thant, Aung, Htay, Htwe and Yar (Thant, et al., 2017)	Assigning Polarity Scores to Facebook Myanmar Movie Comments	2017

Different kinds of machine learning techniques were proposed in Swe and Tin, 2005; Aung and Thein, 2011; Wai, 2011; Wai and Thein, 2011; Win, 2011; Zin, et al., 2011, as listed in Table 2.3. Swe and Tin (2005) developed an optical character recognition system that recognizes characters to identify Myanmar printed words and the translation of Myanmar printed text into the user's own language for many other people who do not understand the Myanmar language in Myanmar to understand Myanmar words. Aung and Thein (2011) proposed an approach to tackle the ambiguity of Myanmar words for Myanmar-English machine translation by disambiguating ambiguous words with part-of-speech. Wai (2011) presented to solve ambiguous verb problems in Myanmar-English translations. Wai and Thein (2011) focused on reordering model for word orders during language translation for the languages which have different word

orders with the Burmese language. It emphasized on to generate reordering rules and to implement Markov-based reordering model which can be incorporated into an English-Myanmar translation model. Win (2011) implemented an effective machine translation system for Myanmar to the English language. The system generates the English sentence by reassembling English words by using English grammar rules. Zin, et al. (2011) presented a translation model based on syntactic structure and morphology of Myanmar language and implemented a subsystem of Myanmar to English translation system.

Table 2.3 Machine Translation

Authors	Title of the paper/project	Year
Swe and Tin (Swe and Tin, 2005)	Recognition and Translation of the Myanmar Printed Text Based on Hopfield Neural Network	2005
Aung and Thein (Aung and Thein, 2011)	Word Sense Disambiguation System for Myanmar Word in Support of Myanmar-English Machine Translation	2011
Wai (Wai, 2011)	Myanmar to English Verb Translation Disambiguation Approach Based on Naïve Bayesian Classifier	2011
Wai and Thein (Wai and Thein, 2011)	Markov-based Reordering Model for English-Myanmar Translation	2011
Win (Win, 2011)	Words to Phrase Reordering Machine Translation System in Myanmar-English Using English Grammar Rules	2011
Zin, Soe, and Thein (Zin, et al., 2011)	Translation Model of Myanmar Phrases for Statistical Machine Translation	2011

The other tasks are processing of speech recognition (Soe and Theins, 2015), implementing search engine (Mon and Mikami, 2011), construction of Myanmar WordNet (Phyue, 2011) and developing speller checker (Mon and Thein, 2013). These papers are listed in Table 2.4.

Table 2.4 Different Burmese Language Processing

Speech Recognition		**Year**
Soe and Theins (Soe and Theins, 2015)	Syllable-based Myanmar Language Model for Speech Recognition	2015
Search Engine		**Year**
Mon and Mikami (Mon and Mikami, 2011)	Myanmar Language Search Engine	2011
Myanmar WordNet		**Year**
Phyue (Phyue, 2011)	Construction of Myanmar WordNet Lexical Database	2011
Spell Checker		**Year**
Mon and Thein (Mon and Thein, 2013)	Myanmar Spell Checker	2013

2.2.7 Low Resource Language

Low resource languages are the languages which lack large monolingual or parallel corpora and/or manually crafted linguistic resources sufficient for building statistical NLP applications (Tesvetkov, 2017). Cieri, et al. (2016) defined low resource languages as the languages "for which few online resources exist" or "for which few computational data resources exist." The Burmese language is one of the low resource languages.

2.3 Word Segmentation

Word segmentation is the process of segmenting the sentence or input text into words. For the languages which have no explicit word boundaries in a written sentence such as Burmese, Thai, Chinese or Khmer, the process of word segmentation has some difficulties. Thus, segmenting words in sentences is one of the indispensable tasks for them.

Word segmentation is required as one of the data pre-processing steps, and it is necessary for NLP research. If this step has many errors, high accuracy of a natural language task may not be expected. For English and many other European languages, it is easy to make word

segmentation if words are separated by a space or an easily noticeable punctuation. They can use tokenization methods or available tools, splitting the text into words. Therefore, the performance of further processing such as part-of-speech tagging, sentiment dictionary construction, sentiment classification and other language processing tasks can give a better result for those kinds of languages. Many works and applications have been proposed in different areas. Thus, a good word segmentation approach is necessary for every language. However, previous proposals of word segmentation of the Burmese language have achieved a lower performance compared to the current state of the art of other languages.

2.3.1 Typical Word Segmentation Methods

The process of word segmentation has usually been conducted in several ways such as rules-based, dictionary-based, statistical, and machine learning approaches by using existing tools and a language model such as n-gram language models. N-gram language model assumes that the probabilistic distribution of each unit is solely based on (n-1) previous units (Maung and Mikami, 2008). The process of word segmentation can also be treated as a classification task with a machine learning framework. Specifically, according to the properties of the text data, a sequence of labeling task for the input with several standard learning frameworks are performed. For the Burmese word segmentation, some papers are based on dictionary-based, statistical, and machine learning approaches Ding, et al. (2016) and some are based on the combination of syllable segmentation and syllable merging (Thet, et al., 2008).

For Thai, which has a writing method similar to the Burmese language, character clustering is proposed (Theeramunkong, et al., 2000; Tongtep and Theeramunkong, 2010). In both languages, the words are formed by the combination of consonants, vowels and tones. Characters are attached in front, back, upper and below of the consonants. In addition, characters are located in three different positions: upper, middle and lower levels. The original concept of Thai Character Cluster (TCC), a unit smaller than a word but larger than a character, was proposed in Theeramunkong, et al. (2000). By applying this concept, 26 types of TCC (TCCT), identifying the type of each character cluster, were proposed to improve Thai word extraction and named entity recognition (Tongtep and Theeramunkong, 2010).

Du, et al. (2016) proposed Chinese word segmentation based on Conditional Random Field (CRF) with character clustering for short text. In their study, characters are grouped by training the Skip-gram model, grouping similar characters. From this model, each character is represented as an embedding vector and these vectors are clustered using K-means algorithm. The

model produces the clusters of character embedding. The result of clustering is used as features to train CRF model for word segmentation.

Chea, et al. (2015) proposed Khmer word segmentation. For the Khmer language, the vocabulary consists of single words and compound words. A single word is a word that is not analyzed into more than one word. When two or more single words, prefixes or/and suffixes are combined, it becomes a compound word. Khmer word segmentation model is trained based on these four patterns of words including single words, compound words, compound words with prefix and compound words with suffix.

2.3.2 Different Proposed Approaches for Burmese Word Segmentation

Ding, et al. (2016) compared the word segmentation for the Burmese language in different ways. In the case of the dictionary-based approach, maximum matching is performed by citing a prepared dictionary. Matching the longest substring in an input sentence is a classic word segmentation. The matching process can be conducted from the beginning of a sentence to its end called forward maximum matching or in reverse direction called backward or reverse maximum matching. They tested these two ways for matching for comparison and decide the better one. Statistical approach model for word segmentation uses the probabilities of words in real textual data. Segmentation results are usually better when the data contain more common words than when the data contain more obscure, less frequent, and/or genre specific words. They employed an N-gram language model which processes the segmentation task by searching the highest probability with the model. In a machine learning framework, a sequence labeling task for each word, syllable, or characters, based on the properties of textual data, is necessary. Among machine learning approaches, CRF in the CRF++ toolkit and SVM in the KyTea toolkit are used for sequential labeling. The output tag-set is binary whether to insert a word boundary or not after each syllable. In their study, the performance of statistical and machine learning approaches is significantly better than that of dictionary-based approaches.

The development of a Myanmar word segmentation method was reported by using Unicode standard encoding (Thet, et al., 2008). Their proposed method is composed of two phases: syllable segmentation by using a rule-based heuristic approach and syllable merging with a dictionary-based approach. For the first step, syllable segmentation, six kinds of heuristic rules are proposed. At the later step, the segmented syllables are merged into words. The input text of segmented syllables is divided into sentences and phrases based on the punctuation marks and spaces. All possible combinations of merged words are generated for each sentence or phrase by

matching the word entries in the dictionary. Then, choose the one with the minimum number of merged words and it is selected as the correctly merged words for the sentence or phrase. However, this way is biased to prefer longer word matching with the dictionary. If two or more combinations have the same minimum number of merged words, the statistical approach is applied to solve this problem. For the statistical approach, the mutual information of two syllables, the probability of observing two syllables together, is pre-calculated with the corpus. Then, it is used to calculate the collocation strength of a sentence or phrase, which is the sum of the collocation strengths of all the merged words in the sentence or phrase. The collocation strength of each word is the sum of the positive strength minus the sum of the negative strength. From all the combinations, the combination with the highest collocation strength is taken. Since the syllable merging process based on the dictionary, there is a problem when the words are unable to merge in a correct way if they are not contained in the dictionary. In addition, if the spelling is incorrect in the test documents or the dictionary, there may be a problem. To solve this problem, the error entries in the dictionary needs to be corrected. The next problem occurs when the segmented syllables are matched with various entries in the dictionary. This problem can be solved by the statistical approach. Their proposed approach considered the characteristics of the Myanmar language and script very well, and the performance can be compared to other language word segmentation tasks.

In this work Pa, et al. (2015), word segmentation is done based on the character segmented data and syllable-segmented data using Condition Random Fields (CRF). The character segmentation segments the sentence into a sequence of graphemes which are represented by the Unicode characters. For syllable segmentation, they applied three general rules. The first rule segments in front of consonants, independent vowels, symbol characters, and numbers. As a second rule, the word breaks in front of subscript consonants, Kinzi characters, and consonant + Asat characters are removed. The third rule is to break the words for special cases such as syllable combinations of loan words (spelling according to the pronunciation of the foreign words) and Pali words. CRF is used to train for two different segmentations of Myanmar: character and syllable. They compared their method with maximum matching, one popular structural segmentation algorithm, as a baseline method. Maximum matching algorithm segments the sentence by matching the longest possible segments in a dictionary. The performance of their approach can be compared with a baseline method.

2.4 Conditional Random Field (CRF)

2.4.1 What is CRF?

Conditional Random Field, CRF, is a machine learning framework for building probabilistic undirected graphical models to segment and label the structured data. Lafferty, Mccallum, and Pereira (2001) reported the performance of CRF outperforms the previous models on natural language data.

2.4.2 How CRF model is trained?

The sequence of input data to the CRF is structured. CRF learns the structured data and constructs a conditional model. A conditional model specifies the probabilities of possible label sequence of input data based on the observation from the sequence of input data. The probability of a transition between labels may depend both on the current observation, and on the past and future observations if the data is available. The observation is based on the feature set which is defined to train the model. CRF model is trained by using maximum likelihood or maximum a posteriori (MAP) estimation to assign a well-defined probability distribution over the possible labels.

The sequence labeling of input data assigns one label for each sequence of the input data. For the word segmentation task, $X=(x_1, x_2, .., x_T)$ as the sequences of characters to be labeled in a sentence with the length of T and $Y=(y_1, y_2, ..., y_T)$ as the corresponding label sequences of each character are given. X is a random variable over data sequences to be labeled which range over the input text and Y is a random variable over the possible labels for input data. A conditional model $p(Y|X)$ is constructed from paired observation and label sequences. Therefore, a CRF is a random field conditioned on the observed input data. A set of feature functions $f=\{f_1, f_2, ..., f_K\}$ are defined which are based on the data and neighboring labels to build a conditional random field model. Weight, λ_i, defining the relationship between each y_i and x_i, is assigned to each function and add all their weight and calculate the probability of each label. $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_K\}$ can be calculated from the observed dataset for a maximum likelihood. CRF is a log-linear model for sequential labels and the form of CRF probabilities is as follows:

$$P(y|x) = \frac{exp(\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(y_{t-1}, y_t, x, t))}{\sum_{y} exp(\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(y_{t-1}, y_t, x, t))} \quad (2.1)$$

Finding the likely sequence of data in word segmentation task is similar to the following equation, Viterbi algorithm, since it searches for the most plausible sequence of node Y* given observed X:

$$Y^* = arg\ max_Y P(Y|X) \tag{2.2}$$

In CRFs, the input data is sequential, and the previous context is taken into account when predictions are made for a data point. To model this behavior, a set of feature functions are necessary to be defined. The feature function is defined as follows:

$$f(X, i, l_{i-1}, l_i) \tag{2.3}$$

The feature function, *f*, contains a set of input vectors, *X*, the index *i* of the data point that is predicted, the label of data point *i-1* in *X* and the label of data point *i* in *X*. It is based on the label of the previous word and the current word. The aim of the feature function is to present some kind of characteristic of the sequence of input data points. To build a CRF model, a set of weights is assigned to each feature function. The weight values are initialized as random values and Gradient Descent updates weight values iteratively, with a small step, until the values converge. The update of the training of CRFs is defined with the following equations:

$$\lambda = \lambda + \alpha\ [\sum_{k=1}^{m} F_j(y^k, x^k) + \sum_{k=1}^{m} p(y|x^k, \lambda)\ F_j(y, x^k)] \tag{2.4}$$

$$F_j(y, x) = \sum_{k=1}^{m} f(X, i, y_{i-1}, y_i) \tag{2.5}$$

To build a CRF model, we first define the feature functions, initialize random values of weights, and then apply Gradient Descent iteratively until the weight values converge. There are two main problems associated with CRF. They are how to estimate the parameter weight values by using an observed data and how to find the best pattern $x = \{x_1, x_2, \ldots, x_T\}$ for a given raw data.

2.4.3 Benefit of CRF

In some applications such as part-of-speech tagging, sequence labelling, we want to predict output random variables, $y = \{y_1, y_2, \ldots, y_T\}$ given an observed input data $x = \{x_1, x_2, \ldots, x_T\}$. Each x_s preserves various information about the input word at position *s*, such as its identity, orthographic features such as prefixes and suffixes, membership in domain-specific lexicons, and information in semantic databases such as WordNet. The purpose is to maximize the number of labels y_s that are correctly classified. For this multivariate prediction problem, one main approach is to learn an independent per-position classifier that maps $x \rightarrow y_s$ for each *s*. However, the problem is that the output variables have complex dependencies like the possibility of having similar labels

for neighboring words in a document or neighboring regions in an image is high, or that the output variables may represent a complex structure such as a parse tree, in which the use of grammar rule near the top of the tree can have a large effect on the rest of the tree.

A simple way to represent how output variables depend on each other is provided by graphical models. A graphical model is a kind of probability distributions that factorize according to an underlying graph. Graphical models represent a complex distribution over many variables as a product of local factors on smaller subsets of variables. Then, it is possible to represent how a given factorization of the probability density corresponds to a particular set of conditional independence relationships satisfied by the distribution. This correspondence makes graphical modeling, a powerful framework for representation and inference in multivariate probability distributions, much more convenient. Learning with graphical models has focused on generative models, directed graphical models, that explicitly try to model a joint probability distribution $p(y, x)$ over inputs and outputs. This approach has both advantages and limitations. The dimensionality of x be very large, and the features can include complex dependencies, so the construction of a probability distribution over them is difficult. This kind of modeling can lead to intractable models and it can lead to a low performance if they are ignored.

One solution to this kind of problem is to model the conditional probability distribution $p(y|x)$ directly with an associated graphical structure, which is called CRF. The CRF model is a conditional undirected graphical model and the dependencies among the input variables do not need to be explicitly represented since it can learn by using the rich and global features of the input. The idea of CRF is that defining a conditional probability distribution over label sequences gave a particular observation sequence, rather than a joint distribution over both label and observation sequences. It is a way of combining the advantages of classification, predicting a single discrete class variable given a vector of features, and graphical modeling, and also a way of combining the ability to model multivariate data compactly with the ability to leverage a large number of input features for prediction. The benefit of a conditional model is that the dependencies that involve only variables in x do not contain in the conditional model. Thus an accurate conditional model can have a much simpler structure than a joint model. The overview of CRF model training and tagging for the sequence of data is shown in Figure 2.2.

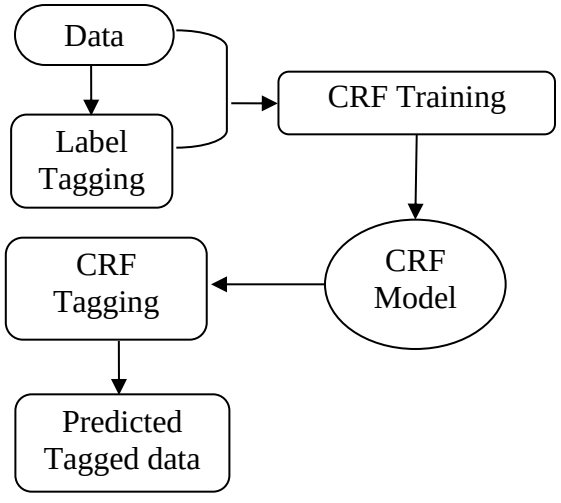


Figure 2.2 Overview of CRF for Segmenting and Labeling Sequence Data

2.5 Distributive Word Representation Models

2.5.1 Word Embedding

In very simplistic terms, Word Embeddings are the texts converted into numbers, and there may be different numerical representations of the same text. (word embeddings or word vectors are numerical representations of contextual similarities between words)

Word embedding is necessary because many Machine Learning algorithms and almost all Deep Learning Architectures are incapable of processing strings or plain text in their raw form. They require numbers as inputs to perform any sort of job, whether it is classification, regression, etc., in broad terms. With the huge amount of data that is present in the text format, it is also imperative to extract knowledge out of it and build applications.

Word representations are a critical component of many natural language processing systems. It is common to represent words as indices in vocabulary, but this fails to capture the rich relational structure of the lexicon. Vector-based models do much better in this regard. They encode continuous similarities between words as distance or angle between word vectors in a high-dimensional space. Similarity in meaning is defined as similarity of vectors:

- Mathematics numbers in vectors should encode meaning
- The environment of a word gives meaning to it
- The more often two words co-occur, the closer their vectors will be

- Two words have close meanings if their local neighborhoods are similar
- Computers can “grasp” the meaning of words by looking at the distance between vectors

The different types of word embeddings can be roughly classified into two categories:

- Frequency-based Embedding
- Prediction-based Embedding

There are two types of embedding: frequency-based embedding and prediction-based embedding.

2.5.2. Frequency-Based Embedding

There are generally three types of vectors that we encounter under this category.

- Feature Vector based on Word Frequency
- TF-IDF Vector
- Co-Occurrence Matrix

2.5.2.1 Feature Vector based on Word Frequency

Term-document or document-term matrix, in which, the rows represent the words and the columns are a bunch of text (e.g., sentences, documents). The cells in the matrix show the frequency of words in each sentence or document. If the words appear many times in some specific datasets, it can be recognized that they are important features for that types of datasets. There may be variations in the way of counting which is taken for each word when constructing the matrix. The counts can be taken as the frequency which is the number of times of word appearances in the data or the presence/absence of the word in the data. Generally, the former one is preferred to the latter one.

2.5.2.2 TF-IDF Vector

Term Frequency-Inverse Document Frequency Matrix, in which, the rows represent the words and the columns are a bunch of text (e.g., sentences, documents). It takes into

account not just the occurrence of a word in a single document but in the entire corpus. The cells represent tf-idf scores. The frequencies of some words, such as pronouns, articles, are higher than some important words in the data. This matrix is to down-weight the common words occurring in almost all documents and give more importance to words that appear in a subset of documents.

$$TF = (\textit{Number of times term t appears in a document}) /$$

$$(\textit{Number of terms in the document})$$

$$IDF = log(N/n) \tag{2.6}$$

where N is the number of documents and n is the number of documents a term t has appeared in.

The idea of IDF is to extract the relevance of the word in a subset of the document. Ideally, if a word appears in almost all the document, then probably that word is not relevant to a specific document. However, if it appears in a subset of documents, then probably there may be some relevance to the documents where it appears.

2.5.2.3 Co-Occurrence Matrix

In the matrix, the rows and columns represent the unique words in the corpus. The cells show the frequency of the two corresponding words co-occurs within a fixed context window. Count vector and tf-idf vector records the frequency or weight of the words, and they do not take into account the relationship between words. The co-occurrence matrix can tell the relationship between words because the words that have similar meaning can appear in the same context. For example, an apple is a fruit, and a mango is also a fruit. Thus, it is expected that the word *apple* and *mango* will appear in similar contexts.

For processing, this co-occurrence matrix is not the word vector representation that is generally used: this is decomposed using techniques like Principal Component Analysis (PCA), Singular Value Decomposition (SVD). The advantage of this matrix is that it preserves the semantic relationship between words. Using SVD which produces more accurate word vector representations than the original matrix and applying factorization which is a well-defined problem and can be efficiently solved. Once the matrix is computed, and it can be used anytime. Because of this fact, it is faster in compared with others. The disadvantage is that it requires huge memory to store the co-occurrence matrix (Analytics Vidhya, 2017; Holzinger, 2016)

2.5.2.4 GloVe Model

GloVe is a popular distributed word representation model. It represents each word with a real-valued vector. The GloVe model learns word vectors by examining word co-occurrence frequency within a text corpus. This is a kind of word frequency-based methods.

GloVe is a new global log-bilinear regression model with a weighted least squares objective. It trains on global word-word co-occurrence counts and combines the advantages of the two major model groups in the literature: global matrix factorization and local context window methods.

The statistics, the frequencies of word appearance, in a corpus is the primary source of available information for all unsupervised methods to learn word representations. Capturing global statistics can be an advantage for the count-based methods. GloVe model utilizes this main benefit of count data while simultaneously capturing the meaningful linear substructures in the vector space. Before training the actual model, we construct a co-occurrence matrix X, where a cell X_{ij} is a "strength" which represents how often the word j appears in the context window of the word i. The probability that word j appears in the context of word i is defined as $P_{ij} = P(j|i) = X_{ij} / X_i$, where $X_i = \sum_k X_{ik}$, the number of appearances of any word in the context of word i. The context window size is pre-defined.

The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus, rather than on the entire sparse matrix or on individual context windows in a large corpus. Populating this matrix requires a single pass through the entire corpus to collect the statistics. For large corpora, this pass can be computationally expensive, but it is a one-time up-front cost. Subsequent training iterations are much faster because the number of non-zero matrix entries is typically much smaller than the total number of words in the corpus.

The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Because the logarithm of a ratio equals the difference of logarithms, this associates the logarithm of ratios of co-occurrence probabilities with vector differences in the word vector space. Because these ratios can encode some form of meaning, this information gets encoded as vector differences as well. Therefore, the word vector learning starts with the ratios of co-occurrence probabilities rather than the probabilities themselves. For this reason, the resulting word vectors perform very well on word analogy tasks.

$$w_i^T \hat{w}_k = log\ (P_{ik}) = log\ (X_{ik} / X_i) = log\ (X_{ik}) - log\ (X_i) \qquad (2.7)$$

where w_i is word vector for main (center) word, and $\hat{w}$ is the separate context word vectors.

Next, define soft constraints for each word pair:

$$w_i^T \hat{w}_k + b_i + b_k = log(X_{ik}) \tag{2.8}$$

Here w_i is the vector for the main word, $\hat{w}_k$ is the vector for the context word, b_i, and b_k are scalar biases for the main and context words. In the above equation, the logarithm diverges whenever its argument is zero. To solve this problem, an additive shift is included in the logarithm, $log(X_{ik})$ changes to $log\ (1 + X_{ik})$, which maintains the sparsity of X while avoiding the divergences. This idea of factorizing the log of the co-occurrence matrix is closely related to matrix factorization method, Latent Semantic Analysis (LSA). The main drawback of this resulting model is that it weighs all co-occurrences equally, even when some words rarely or never co-occur. Those kinds of rare co-occurrences are noisy and carry less information than the more frequent ones.

To address this problem, a new weighted least squares regression model is proposed. Introduce a weighting function $f(X_{ij})$ into the cost function gives the model:

$$J = \sum_{i,j=1}^{V} f(Xij)\ (w_i^T \hat{w}_j + b_i + b_j - log\ X_{ij})^2 \tag{2.9}$$

where, V is the size of the vocabulary. For the weighting function, $f(X_{ij})$, the following function is chosen.

$$f(x) = \begin{cases} (x / x_{max})^{\alpha} & if\ x < x_{max} \\ 1 & otherwise \end{cases} \tag{2.10}$$

There was an observation that the performance could be increased by filtering the data to reduce the effective value of the weighting factor for frequent words (Mikolov et al., 2013). With this idea, the GloVe authors introduce a more general weighting function, which is free to take to depend on the context word as well. The resulting cost function which is equivalent to the previous equation is,

$$\hat{J} = \sum_{i,j} f(X_{ij})\ (w_i^T \hat{w}_j - log\ X_{ij})^2 \tag{2.11}$$

The model generates two sets of word vectors, w and $\hat{w}$. When the matrix, X, is symmetric, w and $\hat{w}$ are equivalent and they vary only as a result of their random initializations, but the two sets of vectors should perform equivalently. In addition, there is evidence that for some types of neural networks, training multiple instances of the network and then combining the results can help to reduce overfitting and noise and generally improve results. With this idea, the sum $w + \hat{w}$ as word vectors are used in the GloVe model. This gives a small boost in performance, with the biggest increase in the processing of semantic analogy task (Pennington, Socher, and Manning,

2014). The idea of applying matrix factorization to the co-occurrence matrix is shown in Figure 2.3.

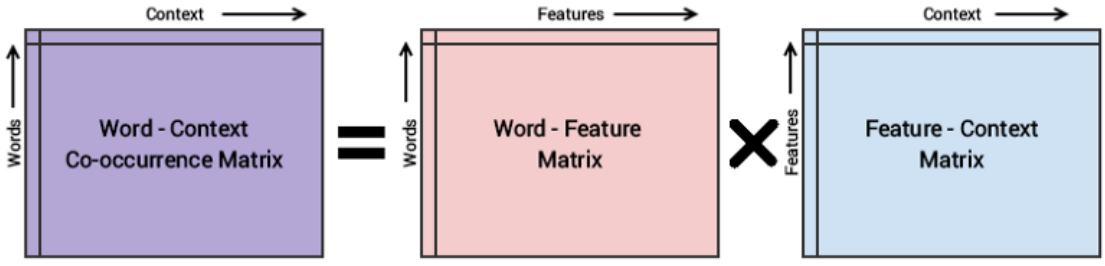


Figure 2.3 Conceptual Model for the GloVe Model's Implementation [1]

2.5.3 Prediction-Based Embedding – Word2vec Model

Word2vec is another kind of distributed word representation model. An efficient representation of the text data is necessary such that conserving information about local word context. This is where the word2vec methodology comes in. The model takes a text corpus as input and produces the word vectors as output. The input text is represented with a one-hot vector. During the training, the model moves through the training corpus with a sliding window. Each instance is a prediction problem: the objective is to predict the current word with the help of its contexts (or vice versa). The model learns output word vectors based on word co-occurrences within a window size. The word vectors represent the words in a numerical way, so-called distributed representations of words. Based on the word vectors, the model can find the words that have a similar meaning. For example, vec(*king*) – vec(*man*) + vec(*woman*) = vec(*queen*). Word2vec training is an unsupervised task, and thus there is no good way to objectively evaluate the result. Evaluation depends on the end application.

Word2vec is a shallow neural network with three layers: an input layer, a hidden layer and an output layer. Two techniques of word2vec model are:

- Continuous bag of words (CBOW)
- Skip-gram

[1] Cited from: https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-glove.html

2.5.3.1 Continuous Bag of Word (CBOW)

The way CBOW works is that it tends to predict the probability of a word given a context. A context may be a single word or a group of words. It is faster and more appropriate for larger corpora. The representation of the CBOW model for a single word in context and the matrix representation are illustrated in Figure 2.4 and Figure 2.5 respectively.

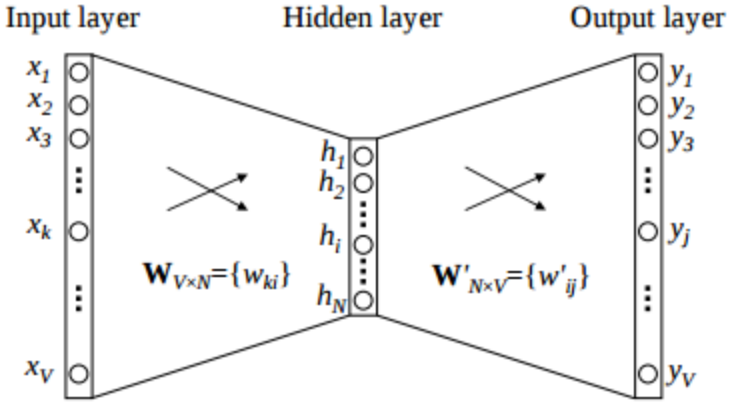


Figure 2.4 Diagrammatic Representation of the CBOW Model (Single Context Word)[2]

The workflow is as follow:

1) Both the input layer and the target have one hot encoded of size [*1 x V*], where *V* is the vocabulary size.
2) There are two sets of weights. The first one is between the input and the hidden layer and the second is between the hidden and the output layer. Input-hidden layer matrix has the size [*V x N*], hidden-output layer matrix size is [*N x V*], where *N* is the number of dimensions to represent a word, and it is also the number of neurons in the hidden layer.
3) This network does not have any activation between layers.
4) The input is multiplied by the input-hidden weights which are called hidden activation.
5) The hidden input is multiplied with hidden-output weights and from which output is calculated.

[2] Cited from: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec

6) The error value between output and target is calculated and propagated backward to re-adjust the weights.
7) Finally, the weight between the hidden layer and the output layer is taken as the word vector representation of the word.

These steps are for a single context word.

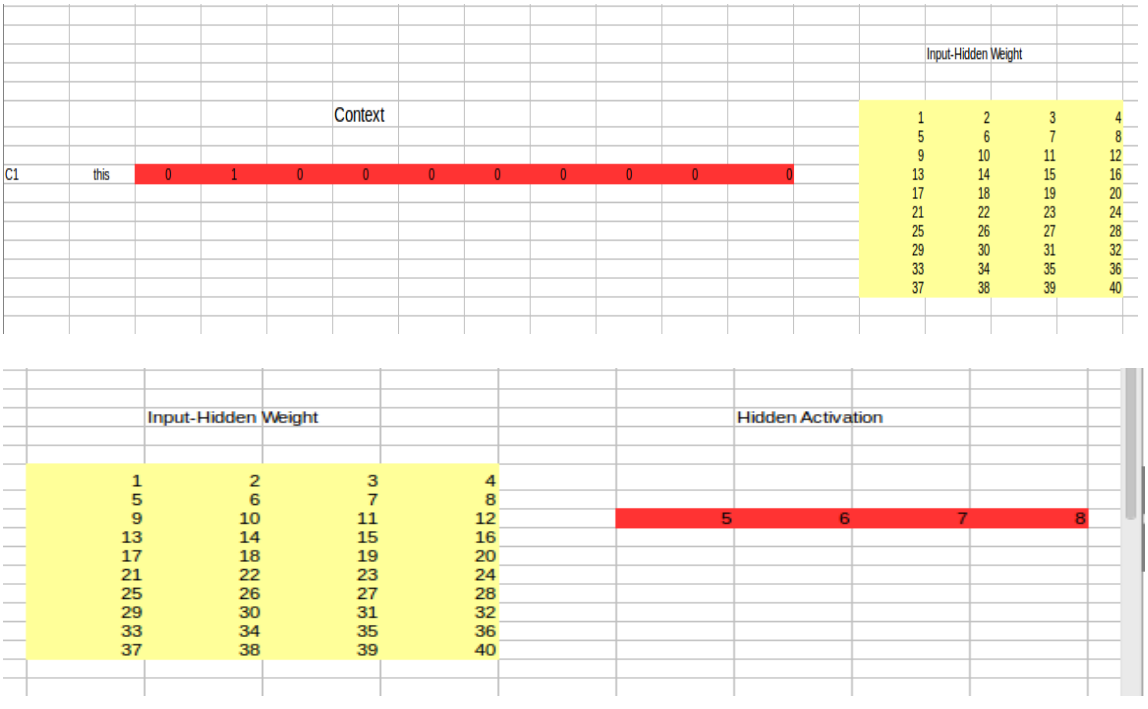


Figure 2.5 Matrix Representation of the CBOW Model (Single Context Word)[3]

The architecture for multiple words in a context and the matrix representation is shown in Figure 2.6 and Figure 2.7, respectively.

[3] Cited from: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec

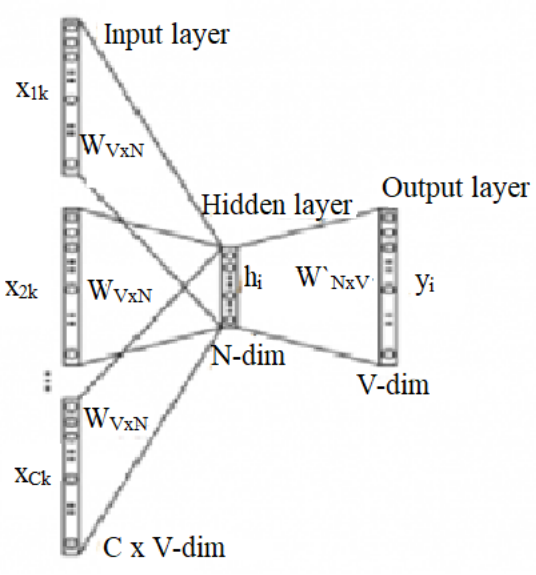


Figure 2.6 Architecture for Multiple Context Words[4]

Context

C1	this	0	1	0	0	0	0	0	0	0	0
C2	corpus	0	0	0	0	1	0	0	0	0	0
C3	context	0	0	0	0	0	0	0	0	1	0

Input-Hidden Weight

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40

Hidden Activation

5	6	7	8
17	18	19	20
33	34	35	36

Average hidden Activation

18.33333333	19.33333333	20.33333333	21.33333333

Figure 2.7 Matrix Representation of the CBOW Model (Multiple Context Words)[5]

[4] Cited from: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec
[5] Cited from: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec

- In this case, an average is taken over all the corresponding rows of the matrix instead of using the corresponding rows of the input-hidden weight matrix to the hidden layer.

The advantage of CBOW is that it runs with a smaller memory size. The disadvantage is that since it takes the average of the context of a word during training, it puts the word between the different clusters of groups. For example, apple can appear between the clusters for fruit and company. In addition, training can last forever if the model is not properly optimized.

2.5.3.2 Skip-gram Model

Skip-gram follows the same architecture as of CBOW, except for its beginning. The objective of skip-gram is to predict the context given a current word. This can produce better word vectors for frequent words, but training is slow. The architecture of the skip-gram model and the matrix representation are shown in Figure 2.8 and Figure 2.9, respectively.

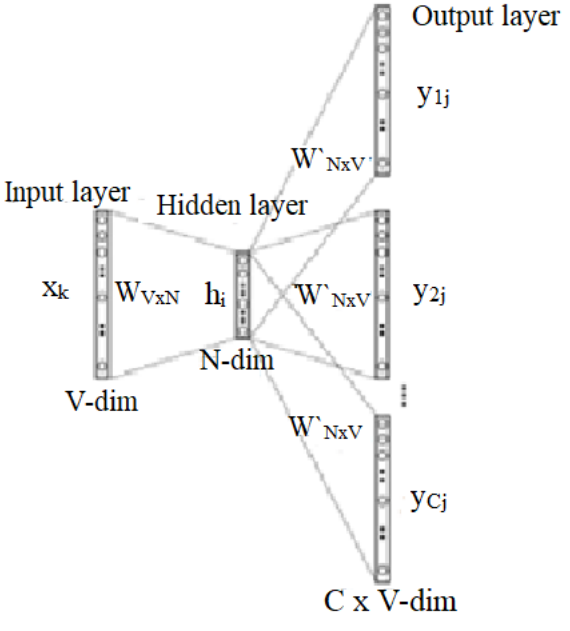


Figure 2.8 Diagrammatic Representation of Skip-gram Model[6]

- After training, the word vector representation takes the weights between the input and the hidden layer. $x_k\, y_{1j}\, W_{VxN}\, W`_{NxV}\, h_i$

[6] Cited from: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec

The advantage of the skip-gram model is that it can capture different semantics for a single word. For example, ‘apple’ has two meanings, a kind of fruit and a company. Generally, skip-gram with negative sub-sampling gives a better result than the other methods.

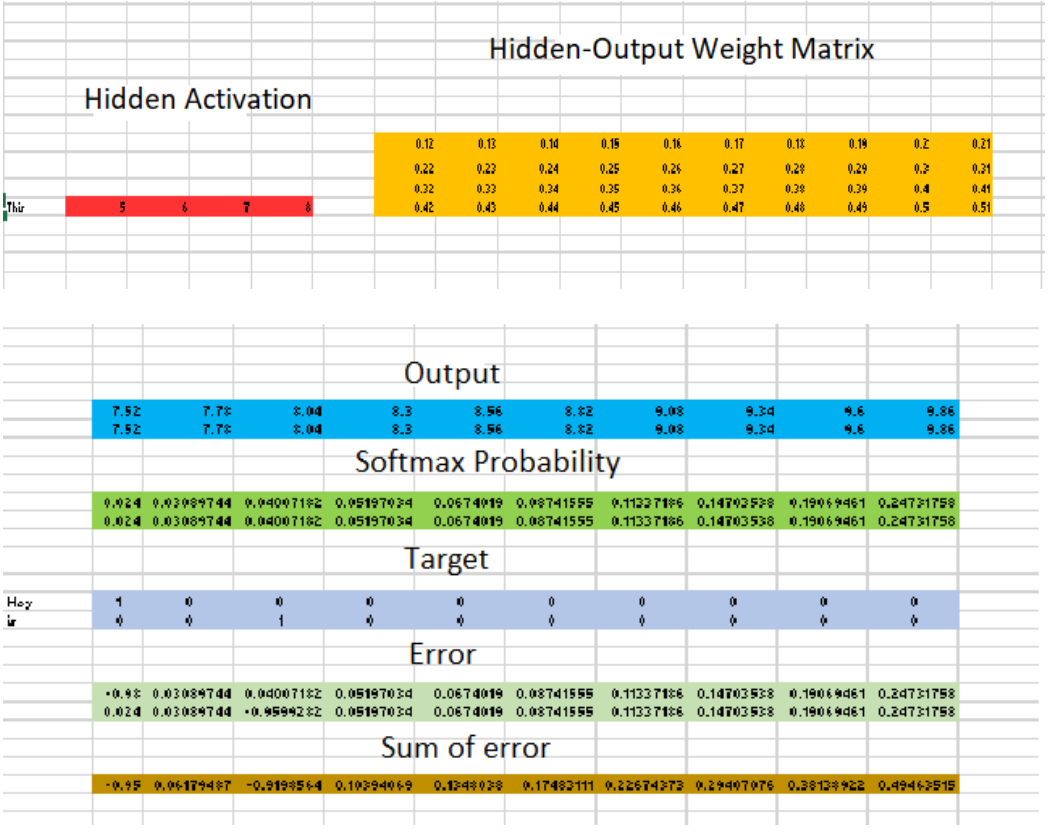


Figure 2.9 Matrix Representation of Skip-gram Model[7]

2.5.3.3 Training Techniques

The basic skip-gram formulation defines $p(w_{t+j} \mid w_t)$ using the softmax function:

$$p\,(w_O \mid w_I) = exp\,(v`_{wO}{}^{T}\, v_{wI}) \,/\, \sum_{w=1}^{W} exp\,(v`_{w}{}^{T}\, v_{wI}) \qquad (2.12)$$

where v_w and $v`_w$ are the input and output vector representations of w, and W is the number of words in the vocabulary. This formulation is impractical because of the cost of computing $\nabla log\ p\ (w_O \mid$

[7] Cited from: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2vec

w_I) proportional to W, which is often large (10^5 – 10^7 terms). The performance of softmax evaluations and updating the weights is slow when the output/vocabulary size is over 10,000.

The softmax function will predict the words which have the highest probability of being in the context of the input word. However, to determine that probability the denominator of the softmax function has to evaluate all the possible context words in the vocabulary. Therefore, if the dimension size for each word vector is 300, 300 × 10,000 = 3M weights need to be trained for the softmax output.

The softmax-based word embedding training approach results in extremely slow training of embedding layers when the data has large word vocabularies. To solve this problem, techniques such as Hierarchical Softmax, Noise Contrastive Estimation and Negative Sampling have been proposed.

(1) Hierarchical Softmax (HS)

The Hierarchical Softmax (HS) is a computationally efficient approximation of the full softmax. The main benefit is that it is needed to evaluate only $log_2(W)$ nodes instead of evaluating W output nodes in the neural network when the probability distribution is calculated.

The hierarchical softmax represents the output layer as a binary tree. The total number of words W words in the output layer are represented as the leaves of the binary tree, and each node explicitly represents the relative probabilities of its child nodes. This representation defines a random walk that assigns probabilities to words.

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left(\left[n(w,j+1) = ch\big(n(w,j)\big) \right] . v`_{n(w,j)}{}^T v_{wI} \right) \qquad (2.9)$$

where, $\sigma(x) = 1 / (1+exp(-x))$. Each word w can be reached by an appropriate path from the root of the tree, $n(w,j)$ is the j-th node on the path from the root to w, $L(w)$ is the length of this path and $ch(n)$ is an arbitrarily fixed child of n, for every inner node n. The cost of computing $log\ p\ (w_O \mid w_I)$ and $\nabla log\ p\ (w_O \mid w_I)$ is proportional to $L(w_O)$, in which the average is not larger than $log\ W$ because it can be verified that $\sum_{w=1}^{W} p(w|w_I) = 1$.

The difference between the standard softmax function of the skip-gram and the hierarchical softmax is that the former one assigns two representations v_w and $v`_w$ for each word w and the latter one has one representation v_w for each word w and one representation

$v`_n$ for every inner node n in the binary tree. The binary tree structure of the hierarchical softmax produces an extent effect on the performance.

(2) Noise Contrastive Estimation (NCE)

There is an alternative, faster method called Noise Contrastive Estimation (NCE). NCE assumes that a good model should be able to differentiate data from noise using logistic regression. Instead of taking the probability of the context word compared to all possible context words in the vocabulary, this method randomly samples 2 to 20 possible context words and evaluates the probability only from these. NCE approximately maximizes the log probability of the softmax. However, this idea is not important for learning word representations. The task of skip-gram is only concerned with learning high-quality vector representations. Therefore, NCE is simplified as long as the vector representations preserve their quality.

(3) Negative Sampling (NEG)

$$log\ \sigma(v`_{wO}{}^T v_{wI}) + \sum_{i=1}^{k} E_{wi \sim Pn(w)} [\ log\ \sigma(-v`_{wi}{}^T v_{wI})] \qquad (2.13)$$

which is used to replace every $log\ P\ (w_O \mid w_I)$ term in the skip-gram. By using logistic regression, the target word w_O is distinguished from draws from the noise distribution $P_n(w)$, in which there are k negative samples for each data sample. The basic idea of negative sampling is to dispense with most of the non-target words from cost function since it is almost the size of entire vocabulary. It works by reinforcing the strength of weights which link a target word to its context words, but rather than reducing the value of all those weights which are not in the context. It simply samples a small number of them that are not in the context. These are called the "negative samples."

The main difference between NCE and the Negative sampling is that the first one needs both samples and the numerical probabilities of the noise distribution, while the second one needs only samples. In addition, NCE can be shown to approximately maximizes the log probability of the softmax.

Typically, to sample negative samples, choose the negative samples based on the uniform distribution or a distribution that is based on the probability of word appearance, $p(w) = count(w) / total$. This sampling distribution is called $p_n(w)$. The research shows that if take the initial distribution raises to the power of three quarter, $p_n(w) \sim p(wi)^{0.75}$ works well. If a word appears in the corpus a lot, probably want to get away from it. The word is not part of the context. New cost function:

$$J = - \sum_{o \in context} log\ \sigma(v_o^T w_i) - \sum_{o \in negative\ samples} log\ \sigma(- v_o^T w_i) \quad (2.14)$$

where w represents the input to the hidden matrix, i is the index of the input word, v is the hidden output matrix and o is the index of the output word (Mikolov et al., 2013).

2.6 Sentiment Analysis

In recent years, people express their opinions with text on websites, newspapers, journals, etc. Classification of a text as positive, negative, or neutral is called sentiment classification, or sentiment analysis (Devi, et al., 2016). Sentiment analysis is useful to know the overall impression of a writer, and it gives the opinion or attitude of a writer. So, it is also known as opinion mining. In other words, this is describing how people feel about a particular topic or product. Sentiment analysis or opinion mining processing employ a variety of data mining and natural language processing (NLP) techniques (Godsay, 2015).

Two typical types of methods for sentiment analysis are machine-learning-based and dictionary-based methods. Machine learning is roughly divided into supervised, semi-supervised and unsupervised learning. When sentiment analysis is considered as a classification issue, this must be supervised learning method. The common techniques used in sentiment analysis such as Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machine (SVM) are supervised learning, and they construct a model to show a binary value, whether a text is classified as positive or negative. This approach requires labeled data to train classifiers. The advantage of machine-learning-based methods is their ability to adapt and create trained models for specific purposes and contexts, and their drawback is the availability of labeled data and hence the low applicability of the method on new data. This is because labeling data might be costly or even prohibitive for some tasks. On the other hand, lexical-based (dictionary-based) methods make use of a predefined list of words, where each word is associated with a specific sentiment value. The lexical methods vary according to the context in which they were created (Godsay, 2015). This is a process of aggregating the sentiment value of each word that contains in the document. From this summarization, the sentiment value of a document is automatically calculated. To do this calculation, sentiment dictionary is necessary to store the sentiment values of words, and there is no supervised data. If the data is supervised, it can be adjusted more appropriately with machine learning techniques.

Sentiment analysis can be conducted at different levels: document level, sentence level or aspect/feature level. In document-level classification, the polarity of sentiment is extracted from the entire review, and a whole opinion is classified based on the overall sentiment of the

opinion holder. The goal is to classify a review as positive, negative, or neutral. This works best when a single person writes the document and expresses an opinion/sentiment on a single entity. Sentence level classification involves two steps: Subjectivity classification of a sentence (subjective and objective) and Sentiment classification of subjective sentences (positive and negative). An objective sentence presents some factual information, while a subjective sentence expresses personal feelings, views, emotions, or beliefs. A subjective sentence may contain multiple opinions and subjective and factual clauses. Both the document level classification and sentence level classification are useful. In aspect/feature level classification, the goal is to identify and extract object features that have been commented on by the opinion holder and determine whether the opinion is positive, negative, or neutral. The features that have a similar meaning are grouped, and a feature-based summary of multiple reviews is produced.

Words express various kinds of sentiments that may be positive, negative, strong, or weak. To perform sentiment analysis, it is important to understand the polarity of words and classify sentiments into positive, negative, or neutral. This task can be accomplished through the use of sentiment lexicons. There are different types of sentiment lexicons that have classified words as positive or negative sentiments (Katrekar, 2014).

2.6.1 Machine Learning Techniques

Machine learning approach in sentiment classification relies on the famous machine learning techniques to classify the text data. Machine learning approach is very practical as it is fully automatic and can handle large collections of data. Machine learning based sentiment classification can be divided into three main categories: supervised, unsupervised and semi-supervised learning methods (Aydogan and Akcayol, 2016).

Supervised learning methods depend on the existence of labeled training documents. Supervised learning is a successful solution in classification and has been used for sentiment classification with highly accurate results. Some of the most frequently used supervised classification methods in sentiment analysis are Naïve Bayes (NB), Maximum Entropy (ME), Support Vector Machine (SVM), Artificial Neural Network (ANN) and Decision Tree (DT) classifiers. Some other less commonly used algorithms are Logistic Regression (LR), K-Nearest Neighbor (KNN), Random Forests (RF) and Bayesian Network.

2.6.1.1 Naïve Bayes (NB)

Naïve Bayes (NB) classifier technique is based on the Bayesian theorem. NB is the simplest and the most commonly used classifier. It is easy to build a Naïve Bayes model without complicated iterative parameter estimation. This makes NB classifier particularly useful for very large datasets. In addition, NB classifier often performs well and is widely used for classification because it often gives better performance than more sophisticated classification methods. The idea of NB classifier is that the conditional probabilities of the independent variables are statistically independent. This is called class conditional independence (Sayad, 2010).

$$P(c|x) = P(x|c)\, P(c) / P(x) \tag{2.15}$$

where, $P(c|x)$ is the posterior probability of class (target) given predictor (attribute), $P(c)$ is the prior probability of class, $P(x|c)$ is the likelihood which is the probability of predictor given class and $P(x)$ is the prior probability of predictor. NB classifiers can handle an arbitrary number of variables which are independent each other whether continuous or categorical. If a set of variables, $X = \{x_1, x_2, x_3, ..., x_d\}$, are given, the posterior probability is constructed for the event C_j among a set of possible outcomes $C = \{c_1, c_2, c_3, ..., c_d\}$.

2.6.1.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the most popular and efficient supervised machine learning algorithms (Weston, 2006). The goal of SVM is to find the optimal separating hyperplane which separates the training data and maximizes the margin of the training data. There are several separating hyperplanes which separate the dataset successfully. However, SVM finds the optimal separating hyperplane which is located as far as possible from each category of data points. This kind of hyperplane provides the best performance for classification.

SVM determines linear separators in the search space which can best separate the different classes. It is ideally suited for text data and it is employed in many sentiment analysis research areas. The example of classification with SVM is illustrated in Figure 2.10. There are different kinds of kernel functions such as linear, polynomial, and sigmoid. The performance of these functions may be same, but the linear function is the simplest form even when training data size is increased. The kernel function for linear SVM is:

$$K(X_i, X_j) = X_i \,.\, X_j \tag{2.16}$$

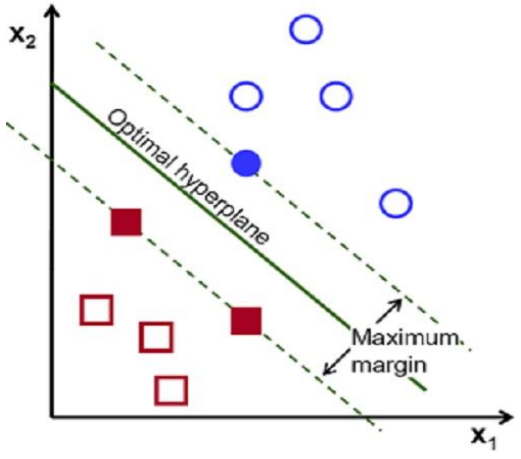


Figure 2.10 Support Vector Machine Classification[8]

2.6.2 Dictionary-Based Methods

Opinion Mining (OM) is analyzing the users' opinions. The polarity lexicon or sentiment dictionary contains the positivity and negativity of a term with their associated numeric sentiment score. The numeric score assigned to each word represents the magnitude of the polarity. These numeric scores can be used for opinion mining task. Most of the researchers use their own sentiment dictionaries and create a dictionary for a particular task. The dictionaries are available for some languages such as English, Chinese, and Japanese. For the Burmese language, however, there is no publicly available sentiment dictionary. In our work, we calculate the sentiment values of each word and create a sentiment dictionary.

There are three strategies to construct a sentiment dictionary or a polarity lexicon: manual, lexicon-based, and corpus-based. The first strategy, the manual way, takes too much time and effort for manual work. Individuals or a group of experts select and annotate the words manually. The second one, the lexicon-based approach, needs an initial list of seed words as input and expands the list by using some publicly available domain independent lexicons such as WordNet or SentiWordNet. The main drawback of this approach is that the resulted lexicon lacks the content and concepts because it uses the domain-independent lexicons. When processing a specialized data, the final lexicon does not have the necessary data. In addition, if the initial list of seed words is not available for the target data, this approach cannot be applied. The final way, the corpus-based approach can provide sufficient coverage of specialized content because it learns the domain-specific lexicon over a training corpus of labeled data in a specific domain. The same word

[8] Cited from: https://aitrends.com/ai-insider/support-vector-machines-svm-ai-self-driving-cars/

in a different domain may have different polarity scores. Therefore, one possible solution is to modify the polarity of the words by using the corpus-based approach (Asghar, et al., 2015). Many researchers have proposed different methods for constructing a sentiment dictionary. Some of the approaches and facts related to my work and the previously proposed methods for constructing Vietnamese and Burmese languages are in the following.

2.6.2.1 SO-PMI

This paper proposed unsupervised learning algorithm to classify reviews as recommended (thumbs up) or not recommended (thumbs down). A review is classified by calculating the average semantic orientation of the phrases in the review that contain adjectives or adverbs. The semantic orientation value is calculated as the mutual information between the given phrase and the selected seed words.

Their proposed unsupervised learning algorithm takes a written review as the input and produces a classification result as the output. The algorithm includes three main steps. The first step is identifying phrases in the input text that contain adjectives or adverbs by using a part-of-speech tagger. The next step is to estimate the semantic orientation of each extracted phrase. If the phrase has good associations, it has a positive semantic orientation; otherwise, it has a negative semantic orientation. The final step is to assign the input review to a class, recommended or not recommended, according to the average semantic orientation of the extracted phrases from the review. If the average value of the review's phrases is positive, the review is in recommended class. If it is negative, the review is in not recommended class.

To estimate the semantic orientation of a phrase, the PMI-IR algorithm is employed which uses Pointwise Mutual Information (PMI) and Information Retrieval (IR). PMI-IR measures the similarity of pairs of words or phrases. PMI between two words, $word_1$ and $word_2$, is calculated as follows:

$$PMI\ (word_1 \text{ and } word_2) = log_2\ [\ p(word_1\ \&\ word_2)\ /\ p(word_1)\ p(word_2)] \qquad (2.17)$$

Here, $p(word_1\ \&\ word_2)$ is the probability of co-occurrence of $word_1$ and $word_2$. The ratio between $p(word_1\ \&\ word_2)$ and $p(word_1)\ p(word_2)$ is a measure of the statistical dependence degree between the words. Taking the log of this ratio is the amount of information that the presence of one word when the other word is observed. The Semantic Orientation (SO) of a phrase is calculated as follows:

$$SO\ (phrase) = PMI\ (phrase,\ 'excellent') - PMI\ (phrase,\ 'poor') \qquad (2.18)$$

Here, ‘excellent’ and ‘poor’ are selected as the strongest opinion words from the review rating system. Mostly, ‘poor’ is defined in one-star rating and ‘excellent’ by five-star rating. Thus, the value of SO is positive when the given phrase is more strongly associated with the word ‘excellent’ and negative when it is more strongly associated with the word ‘poor’ (Turney, 2002).

2.6.2.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by applying statistical computations to a large corpus of text. The main idea is that the totality of information about all the word contexts in which a given word does and does not appear provides a set of mutual constraints. That can determine and represent the similarity of meaning of words and set of words to each other by statistical analysis of large text corpora. LSA is based on singular value decomposition, a mathematical matrix decomposition technique to factor analysis.

As a practical approach, LSA produces measures of word-word, word-passage and passage-passage relations that are reasonably well correlated with several human cognitive phenomena involving association or semantic similarity. The correlation must be the result of the way that people' representation of meaning is reflected in the word choice of writers, and vice-versa, that people' representations of meaning reflect the statistics of what they have read and heard. With LSA, human judgments of overall meaning similarity can be approximated. However, the similarity estimates derived by LSA are not simple contiguity frequencies or co-occurrence contingencies but depend on a deeper statistical analysis (thus the term is “Latent Semantic”). That is capable of correctly inferring relations beyond the first-order co-occurrence and, as a consequence, it is often a much better predictor of human meaning-based judgments and performance.

LSA differs from other statistical approaches in two significant respects. First, the LSA analysis uses not just the summed contiguous pairwise (or tuple-wise) co-occurrences of words as its initial data, but the detailed patterns of occurrences of words over very large numbers of local meaning-bearing contexts, such as sentences or paragraphs.

Second, the LSA method assumes that the choice of dimensionality in which all of the local word-context relations are jointly represented that reduce the dimensionality (the number of dimensions by which a word or passage is described) of the observed data from the number of initial contexts to a much smaller. Even if it is still a large number, it will often produce much better approximations to human cognitive relations. Thus, an important component of applying the

technique is finding the optimal dimensionality for the final representation. Another fact, unlike many other methods, is that LSA employs a pre-processing step in which the overall distribution of words over the whole contexts, independent of their correlations, is taken into account which improves LSA's results considerably (Landauer, Foltz, and Laham, 1998).

(1) Details about Creating LSA Semantic Spaces

LSA is a fully automatic mathematical/statistical technique for extracting and inferring relations of expected contextual usage of words or passages. It is not a traditional natural language processing or artificial intelligence program because it does not use manually constructed dictionaries, knowledge bases, semantic networks, grammars, syntactic parsers, or morphologies and so on. It takes only raw texts as its input and parsed into words defined as unique character strings and separated into meaningful passages or units such as sentences or paragraphs.

The procedure is as follows. The first step is to represent the text as a matrix in which each row stands for a unique word, and each column stands for a text passage or other context. Each cell contains the frequency with which the word of its row appears in the passage denoted by its column. Next, the cell entries are subject to a preliminary transformation in which each cell frequency is weighted by a function that expresses both the word's importance in the particular passage and the degree to which the word type carries information in the domain of discourse in general.

As a next step, LSA applies singular value decomposition (SVD) to the matrix. This is a form of factor analysis or more properly the mathematical generalization of which factor analysis is a special case. The idea of SVD is that to have a dense vector (eliminating as much 0's as possible to keep only the relevant values) with a low number of dimensions. This approach can generate useful semantic and syntactic relationships. In SVD, a rectangular matrix is decomposed into the product of three other matrices.

Among three matrices, one component matrix describes the original row entities as vectors of derived orthogonal factor values, the second one describes the original column entities in the same way, and the third is a diagonal matrix containing scaling values. When the three components are matrix-multiplied, the original matrix is reconstructed. There is a mathematical proof that any matrix can be decomposed perfectly, without using more factors than the smallest dimension of the original matrix. When the number of the factors used is smaller than the necessary number, the reconstructed matrix is a least-squares best fit. One can reduce the dimensionality of the solution simply by deleting coefficients in the diagonal matrix, ordinarily

starting with the smallest. Practically, for computational reasons, only a limited number of dimensions can be constructed for very large corpora (Landauer, Foltz, and Laham, 1998).

$X = UDV^T$, where U and V are orthogonal matrices and D is a diagonal matrix of singular values. The compressed version of original matrix with k singular values is:

$$X = U_k D_k V_k^T \tag{2.19}$$

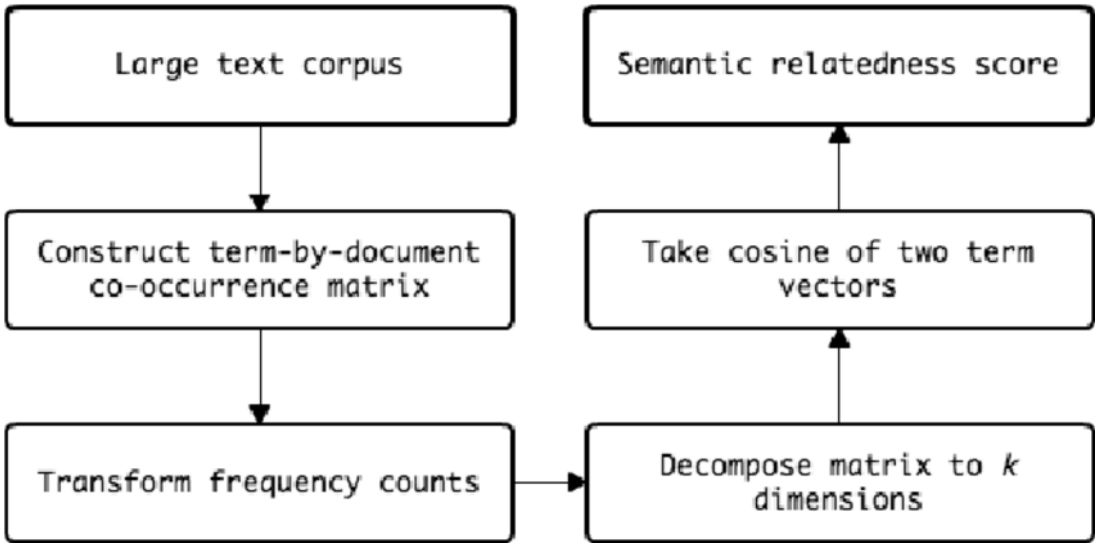


Figure 2.11 Latent Semantic Analysis[9]

(2) Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a way for factorizing a matrix into singular vectors and singular values. Every real matrix has a singular value decomposition. The matrix is a product of three matrices:

$$A = U\,D\,V^T \tag{2.20}$$

For example, A is an $m \times n$ original matrix, and then, U is an $m \times m$ matrix, D is an $m \times n$ matrix, and V is an $n \times n$ matrix. Each of these three matrices (U, D and V) has a special structure. The matrices U and V are orthogonal matrices, and the matrix D is a diagonal matrix. The diagonal elements of D are the singular values of the original matrix A, the column elements of U are the left-singular vectors, and the column elements of V are the right-singular vectors. When these three matrices are combined, the original matrix A is obtained. SVD performs the dimension reduction

[9] Cited from: https://www.researchgate.net/figure/292608200_fig5_Figure-5-Steps-in-latent-semantic-analysis

and preserves the important information from the original data (Goodfellow et al., 2016). The idea of factorizing a matrix is shown in Figure 2.10.

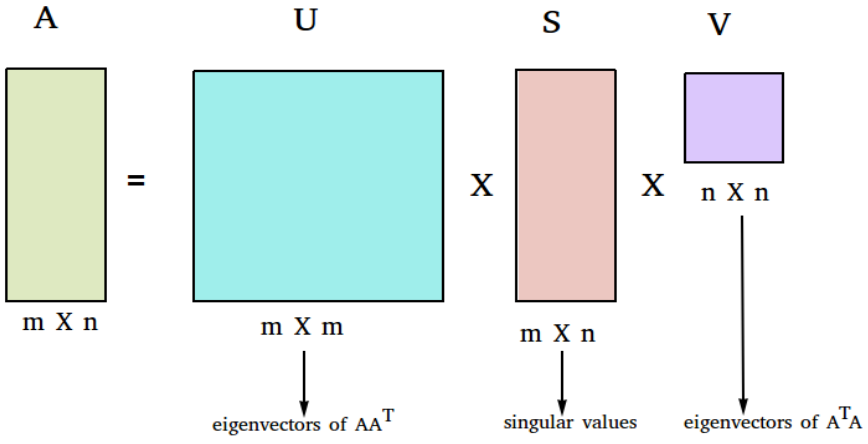


Figure 2.12 Singular Value Decomposition[10]

2.6.2.3 Semantic Orientation-Pointwise Mutual Information (SO-PMI) and Semantic Orientation-Latent Semantic Analysis (SO-LSA)

The evaluative character of a word is called its semantic orientation. Positive semantic orientation indicates positive opinion like praise and negative semantic orientation indicates negative opinion like criticism. The semantic orientation can be varied in two directions: positive or negative and degree: mild or strong. Turney and Littman (2003) introduced the calculation of SO of a word from its associated statistical information with a set of positive and negative seed words. The set of seed words is carefully chosen instead of randomly selected before processing. If the value of semantic orientation of a word is positive, the word is positive. Otherwise, it is negative. This approach is evaluated in two ways based on two different statistical measures of word association. One is Pointwise Mutual Information (PMI), which measures the association degree between two words by the frequency of their co-occurrence, and the other is Latent Semantic Analysis (LSA), which measures the association degree of two words by comparing the contexts in which they co-occur.

SO-PMI is based on the following equations:

$$PMI\ (word_1, word_2) = log_2\ [\ p(word_1\ \&\ word_2)\ /\ p(word_1)\ p(word_2)] \quad (2.21)$$

$$SO\text{-}PMI\ (word) = \sum_{pword\ \in Pwords} PMI\ (word, pword) - \sum_{nword\ \in Nwords} PMI\ (word, nword) \quad (2.22)$$

SO-LSA is based on the following equations:

[10] Cited from: https://blog.paperspace.com/dimension-reduction-with-principal-component-analysis

$$SO\text{-}LSA(word) = \sum_{pword \in Pwords} LSA(word, pword) - \sum_{nword \in Nwords} LSA(word, nword) \quad (2.23)$$

Here, the calculation of LSA includes two main steps. The first step is to construct a matrix for the input data. The next step is to apply singular value decomposition to compress the original matrix. LSA measures the similarity of words by using the compressed matrix instead of original matrix.

In the above equations, Pwords means the set of positive seed words and Nwords means the set of negative seed words. When SO-PMI of a word is positive, the word is classified as having a positive semantic orientation, and it has a negative orientation when SO-PMI is negative. The magnitude represents the strength of the semantic orientation of a word. The classification with SO-LSA is the same condition with SO-PMI. In their experiments, three different corpora are used for unsupervised learning, and two different lexicons are used to evaluate the results of the learning. Their experiments suggested that SO-LSA method used the data more efficiently than SO-PMI method and SO-LSA might provide better accuracy than SO-PMI for a corpus of comparable size (Turney, and Littman, 2003).

(1) Cosine Similarity

There are many ways to compute the similarity of words or other units. Cosine similarity is one of the popular ways. It is a measure that calculates the cosine of the angle between two vectors, where each vector may represent word or document. Cosine Similarity generates a metric that expresses how two vectors are related by looking at the angle instead of magnitude. Therefore, this matrix is a measurement of orientation, not in magnitude. The cosine similarity formula is as follows:

$$cos\ \theta = \frac{\vec{a} \,.\, \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} \quad (2.24)$$

An angle of 0° means that cos θ = 1 and that the vectors have identical directions, i.e., the corresponding two vectors are completely similar to one another. Otherwise, an angle of 90° means that cos θ = 0 and that the corresponding vectors are perpendicular, but not necessarily that are uncorrelated (Garcia, 2015).

2.6.2.4 SO and SM + SO

The basic assumption of identifying the sentiment terms is that the terms that have similar orientation tend to co-occur at the document level. This paper proposed a second assumption

that the sentiment terms that have opposite orientation tend not to co-occur at the sentence level. In their work, they chose four positive and six negative seed words manually which tend to be the strongest bearers of sentiment and frequently appear in their data. The seed words are used in two basic ways. The first way is to determine the semantic orientation of words in the corpus called semantic orientation or SO method. This way is based on the assumption that the words which have similar orientation tend to co-occur. The second one is to mine sentiment vocabulary from the unlabelled data with the additional assumption that the words which have opposite orientation tend not to co-occur. This is called sentiment mining or SM method which produces a set of sentiment words but no orientation for that set of terms. To find the semantic orientation for this set of terms, SO method is used.

The procedure of SO method is as follows:

1) *PosScore* (*f*) = *PosScore* (*f*) + *PMI* (*f, spos*)
   NegScore (*f*) = *NegScore* (*f*) + *PMI* (*f, sneg*)
2) *PosScore* (*f*) = *PosScore* (*f*) */ number of positive seed words*
   NegScore (*f*) = *NegScore* (*f*) */ number of negative seed words*
3) *SO* = *PosScore* (*f*) – *NegScore* (*f*)

Here, *f* means feature in the data, spos and sneg mean positive and negative seed words respectively.

The procedure of SM + SO method is as follows: the PMI of each feature from the data and each seed word from the seed words list is calculated. The lowest score concerning any of the seed words is recorded as a score for each feature. After choosing the lowest PMI score for all features, the features that range over the top n% of features that have lowest scores are identified. The different values for n are tested to get the better choice of words. To find the semantic orientation for this set of features, SO method is applied.

The result from these two methods SO and SM+SO is a list of features with their associated semantic orientation scores. The higher the score, the more positive orientation and vice versa. This list can be used for classification for the input data. The data are classified by adding the scores of all features that are contained in the data. In their experiment, the distribution of class labels in the data is 15.5%, 21.5% and 63.0% for negative, neutral and positive sentences respectively. For the final classification, the scores of all sentences are sorted, and the thresholds are determined as a classifier ratio. The top 63.0% and bottom 15.5% of scores for positive and negative respectively (Gamon and Aue, 2005).

2.6.2.5 Construction of a Sentiment Dictionary: a case of Vietnamese

Few attempts have been made to construct a sentiment dictionary for low resource languages. One of the attempts was made for Vietnamese (Vu and Park, 2014). They proposed an approach to construct a Vietnamese SentiWordNet (VSWN) from Vdict, which is a deliberately constructed dictionary and is useful for their work. It consists of 39,561 words in which each word has essential information about morphology, syntactic, and semantic. This information can provide coverage to derive synset information. There are two main steps in VSWN construction: building VSWN core phase and semi-supervised learning phase. The first step is to obtain VSWN core form English SentiWordNet (ESWN). The second step is to construct two classifiers by using the VSWN core and calculate opinion-related properties for all the synsets. Synset is a set of words which have the same meaning.

In the building core phase, they contributed a VNComments corpus with lack of Vietnamese lexical resources. The corpus is about the top electric news in Vietnam. First, they used Google Translate API to translate all the synsets which have Pos(s) or Neg(s) above 0.4 in ESWN in order to detect the words that have less ambiguity opinion. Since the quality of Google Translate API is limited, only the coincide terms between VNComments and the translated synsets are selected. Then, five annotators are employed to select the top 1,017 synsets. After that, the VNComments corpus is used to modify Pos(s), Neg(s) of each word in all the synsets by using the following equations:

$$Pos(w) = \#_{pos}(w) / \#(w) \tag{2.25}$$

$$Neg(w) = \#_{neg}(w) / \#(w) \tag{2.26}$$

where $\#_{pos}(w)$ and $\#_{neg}(w)$ are the occurrences of word w in pos set and neg set respectively and $\#(w)$ is the total occurrences of word w in the VNComments corpus.

Since there is no WordNet for the Vietnamese language, they applied semi-supervised learning phase to achieve efficient SentiWordNet. This phase includes four sub-steps. First, instead of WordNet, VSWN core is employed to generate the three seed sets (positive, negative, and neutral) with a heuristic threshold of 0.3. Since VSWN core is limited, expand the three sets by using synonym and antonym relations in Vdict. The next step is to generate positivity and negativity classifiers based on the combination of three expanded sets. Then, all the entries in Vdict are extracted and classified by using two classifiers from the previous step. Each synset has two margins represented positive score and negative score. The final step is to normalize these scores into the interval [0, 1]. The scores for each synset are recalculated as the following equation:

$$f(Øi) = log\ [\ Øi / Max\ (abs(Øp) + abs(Øn))] \tag{2.27}$$

where *Øi* is a margin of synset, *Øp* is positive margin of synset and *Øn* is negative margin of synset. After this calculation, each synset achieves positive and negative scores. According to this way, they investigated a SentiWordNet's construction method to generate VSWN (Vu and Park, 2014).

2.6.2.6 Construction of Myanmar WordNet Lexical Database

This paper proposed a semi-automatic approach to construct Myanmar WordNet lexical database from WordNet which provides lexical and semantic relations among English words, Fellbaum, (1998) and Myanmar English and English-Myanmar Machine Readable Dictionaries (MRDs) which provide the translation relations between Myanmar and English words (Phyue, 2011). Lexical semantic and translation relations can be obtained from WordNet and MRDs. The method includes three steps: the MRD extraction phase, the link analyzing phase and the WordNet construction phase.

The first step, MRD extraction phase, is to extract the lexical information from the available resources. The data from many resources with different format needs to be converted into a common form and joins and manages the scattered data to access smoothly. Then, the data is grouped according to their part-of-speech (POS). The second step, the link analyzing phase, analyzes and classifies the translation links, the relationships between Myanmar and English words from Bilingual MRDs, with respect to semantic links. The semantic links, the relationships between English words and their meaning, are obtained from WordNet. The translation links are verified for the lexical gap between two languages and evaluate the candidate set of links. The candidate links represent the relationships between Myanmar words and their meaning. Finally, from the verified translation links and WordNet, Myanmar WordNet is constructed in which Myanmar words are organized by synsets, a group of words having the same meaning. The third step, construction phase, supports the relationships between words and meanings and attaches the glossary. The relationships between Myanmar and English words from the second step and WordNet are used to attach Myanmar words to English words. In Myanmar WordNet, the words are classified with respect to the synonym sets in WordNet and the glossary can be obtained from the Bilingual MRDs. The transitivity relation equation is employed in this phase:

$$A \boldsymbol{R} B \wedge B \boldsymbol{R} C => A \boldsymbol{R} C \tag{2.28}$$

This transitivity is a mathematical property of binary relations such that if A and B, and B and C are related, then A and C are also related, for all A, B, and C. This property is applied to the relations between words and synsets in English WordNet. If English word in synset1 has internal relation with another word in synset2 and both English words have translation link to Myanmar

words, then this internal relation can infer to Myanmar as well. Therefore, the relations in English WordNet can be inferred to Myanmar internal relation (Phyue, 2011).

2.7 Concluding Remarks

In this chapter, theories and other people's methods that are related to my research are discussed. The main part of my research and how these related works are applied to my research are going to explain in the next chapter.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter explains the details of my research methodology step-by-step. The methodology includes four main steps, namely, word segmentation (in two different ways), words grouping with distributive word models (word2vec and GloVe), sentiment values calculation and finally, sentiment classification. The process of classification is tested in two different ways.

3.2 Overview of Research Methodology

The overview of whole research is shown in Figure 3.1. First, the data is collected and made surveys. Second, the data is segmented in two different ways. The data from the second way of segmentation is too much larger than the first one. Therefore, in the next step, the data is grouped by using two models: word2vec and GloVe comparatively. Then, the extracted data from four different ways are used to calculate sentiment values and create sentiment dictionary. Finally, the process of classification is performed in two different ways.

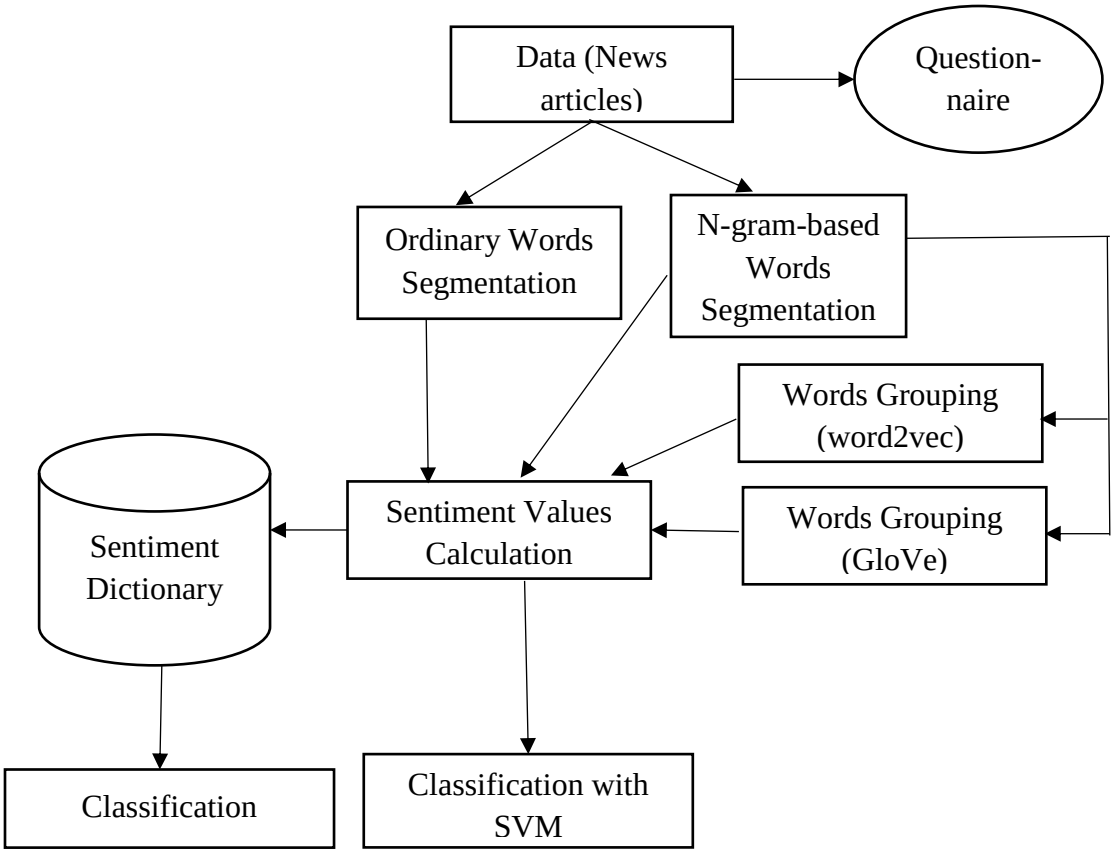


Figure 3.1 Overview of Research Methodology

3.3 Types of Data

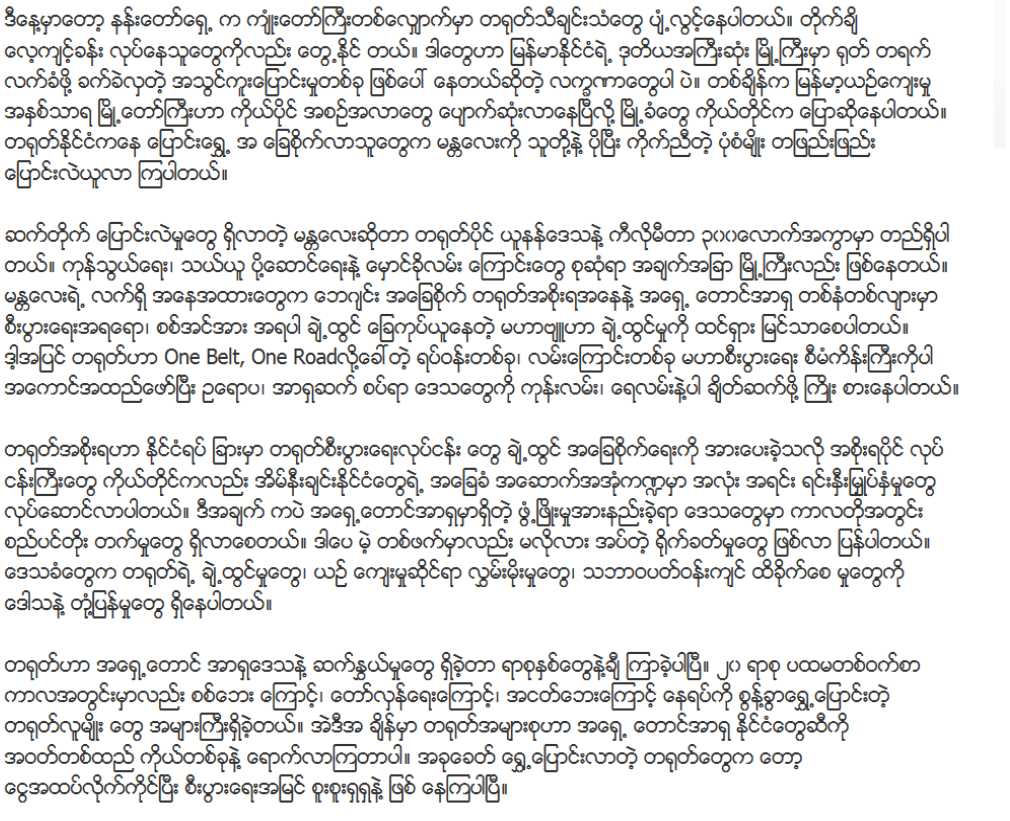
In this research, two types of data are employed. The first data is newspaper articles of 7Day Daily, one of the most popular private newspapers in Myanmar. The daily news of this newspaper can be obtained from its official website (7Day Daily). The news articles are categorized into Election, News, Business, Life, Entertainment, Opinion, Politics, Regional, Sports, World News, Cartoon, Videos and Special Reports on the website. Data is collected only from the Opinion group because this category contains the opinions from many people about the news and that can be used for positive and negative opinion classification. As an example, one news article on the web page is shown in Figure 3.2. Because of the length of the article, only some paragraphs from the actual article are shown in the figure.

Web crawling method is used to collect the data from websites. Web crawling is a technique of downloading web pages associated with given URLs. It extracts hyperlinks contained

in the downloaded pages and also, downloads web pages found in the hyperlinks. The necessary text from the web page can be extracted by specifying the associated parameter values.

Among the collected news articles, I made a survey for 500 news articles to get the opinion of people whether the news is positive or negative. I distributed the surveys to the university students in Mandalay, Myanmar. Every article is read by at least two people. If the result from two people is different, the third person is asked to read that article. The final classification for each article is based on the majority choice. With this survey's result, the performance of my research is evaluated.

The second data is Amazon reviews data. It can be obtained from He, R., and McAuley (2016); McAuley, et al. (2015). Each review has a score from 1 (very negative) to 5 (very positive). It is decided to classify the data that is the reviews which have the score of 1 or 2 are negative reviews and 4 or 5 are positive reviews. For data collection, 10,000 positive reviews and 10,000 negative reviews are collected. The details information of data and questionnaire surveys will be explained in the next chapter.



ဒီနေ့မှာတော့ နန်းတော်ရှေ့ က ကျူးတော်ကြီးတစ်လျှောက်မှာ တရုတ်သီချင်းသံတွေ ပျံ့လွင့်နေပါတယ်။ တိုက်ချီ လေ့ကျင့်ခန်း လုပ်နေသူတွေကိုလည်း တွေ့နိုင် တယ်။ ဒါတွေဟာ မြန်မာနိုင်ငံရဲ့ ဒုတိယအကြီးဆုံး မြို့ကြီးမှာ ရှတ် တရက် လက်ခံဖို့ ခက်ခဲလှတဲ့ အသွင်ကူးပြောင်းမှုတစ်ခု ဖြစ်ပေါ် နေတယ်ဆိုတဲ့ လက္ခဏာတွေပါ ပဲ။ တစ်ချိန်က မြန်မာ့ယဉ်ကျေးမှု အနှစ်သာရ မြို့တော်ကြီးဟာ ကိုယ်ပိုင် အစဉ်အလာတွေ ပျောက်ဆုံးလာနေပြီလို့ မြို့ခံတွေ ကိုယ်တိုင်က ပြောဆိုနေပါတယ်။ တရုတ်နိုင်ငံကနေ ပြောင်းရွှေ့ အ ခြေစိုက်လာသူတွေက မန္တလေးကို သူတို့နဲ့ ပိုပြီး ကိုက်ညီတဲ့ ပုံစံမျိုး တဖြည်းဖြည်း ပြောင်းလဲယူလာ ကြပါတယ်။

ဆက်တိုက် ပြောင်းလဲမှုတွေ ရှိလာတဲ့ မန္တလေးဆိုတာ တရုတ်ပိုင် ယူနန်ဒေသနဲ့ ကီလိုမီတာ ၃၀၀လောက်အကွာမှာ တည်ရှိပါ တယ်။ ကုန်သွယ်ရေး၊ သယ်ယူ ပို့ဆောင်ရေးနဲ့ မှောင်ခိုလမ်း ကြောင်းတွေ စုဆုံရာ အချက်အခြာ မြို့ကြီးလည်း ဖြစ်နေတယ်။ မန္တလေးရဲ့ လက်ရှိ အနေအထားတွေက ဘေဂျင်း အခြေစိုက် တရုတ်အစိုးရအနေနဲ့ အရှေ့ တောင်အာရှ တစ်နံတစ်လျားမှာ စီးပွားရေးအရရော၊ စစ်အင်အား အရပါ ချဲ့ထွင် ခြေကုပ်ယူနေတဲ့ မဟာဗျူဟာ ချဲ့ထွင်မှုကို ထင်ရှား မြင်သာစေပါတယ်။ ဒါ့အပြင် တရုတ်ဟာ One Belt, One Roadလို့ခေါ်တဲ့ ရပ်ဝန်းတစ်ခု၊ လမ်းကြောင်းတစ်ခု မဟာစီးပွားရေး စီမံကိန်းကြီးကိုပါ အကောင်အထည်ဖော်ပြီး ဥရောပ၊ အာရှဆက် စပ်ရာ ဒေသတွေကို ကုန်းလမ်း၊ ရေလမ်းနဲ့ပါ ချိတ်ဆက်ဖို့ ကြိုး စားနေပါတယ်။

တရုတ်အစိုးရဟာ နိုင်ငံရပ် ခြားမှာ တရုတ်စီးပွားရေးလုပ်ငန်း တွေ ချဲ့ထွင် အခြေစိုက်ရေးကို အားပေးခဲ့သလို အစိုးရပိုင် လုပ် ငန်းကြီးတွေ ကိုယ်တိုင်ကလည်း အိမ်နီးချင်းနိုင်ငံတွေရဲ့ အခြေခံ အဆောက်အအုံကဏ္ဍမှာ အလုံး အရင်း ရင်းနှီးမြှုပ်နှံမှုတွေ လုပ်ဆောင်လာပါတယ်။ ဒီအချက် ကပဲ အရှေ့တောင်အာရှမှာရှိတဲ့ ဖွံ့ဖြိုးမှုအားနည်းခဲ့ရာ ဒေသတွေမှာ ကာလတိုအတွင်း စည်ပင်တိုး တက်မှုတွေ ရှိလာစေတယ်။ ဒါပေ မဲ့ တစ်ဖက်မှာလည်း မလိုလား အပ်တဲ့ ရိုက်ခတ်မှုတွေ ဖြစ်လာ ပြန်ပါတယ်။ ဒေသခံတွေက တရုတ်ရဲ့ ချဲ့ထွင်မှုတွေ၊ ယဉ် ကျေးမှုဆိုင်ရာ လွှမ်းမိုးမှုတွေ၊ သဘာဝပတ်ဝန်းကျင် ထိခိုက်စေ မှုတွေကို ဒေါသနဲ့ တုံ့ပြန်မှုတွေ ရှိနေပါတယ်။

တရုတ်ဟာ အရှေ့တောင် အာရှဒေသနဲ့ ဆက်နွယ်မှုတွေ ရှိခဲ့တာ ရာစုနှစ်တွေနဲ့ချီ ကြာခဲ့ပါပြီ။ ၂၀ ရာစု ပထမတစ်ဝက်စာ ကာလအတွင်းမှာလည်း စစ်ဘေး ကြောင့်၊ တော်လှန်ရေးကြောင့်၊ အင်ပါယာကြောင့် နေရပ်ကို စွန့်ခွာရွှေ့ပြောင်းတဲ့ တရုတ်လူမျိုး တွေ အများကြီးရှိခဲ့တယ်။ အဲဒီအ ချိန်မှာ တရုတ်အများစုဟာ အရှေ့ တောင်အာရှ နိုင်ငံတွေဆီကို အဝင်တစ်ထည် ကိုယ်တစ်ခုနဲ့ ရောက်လာကြတာပါ။ အခုခေတ် ရွှေ့ပြောင်းလာတဲ့ တရုတ်တွေက တော့ ငွေအထပ်လိုက်ကိုင်ပြီး စီးပွားရေးအမြင် စူးစူးရှရှနဲ့ ဖြစ် နေကြပါပြီ။

Figure 3.2 Image of The Actual Newspaper

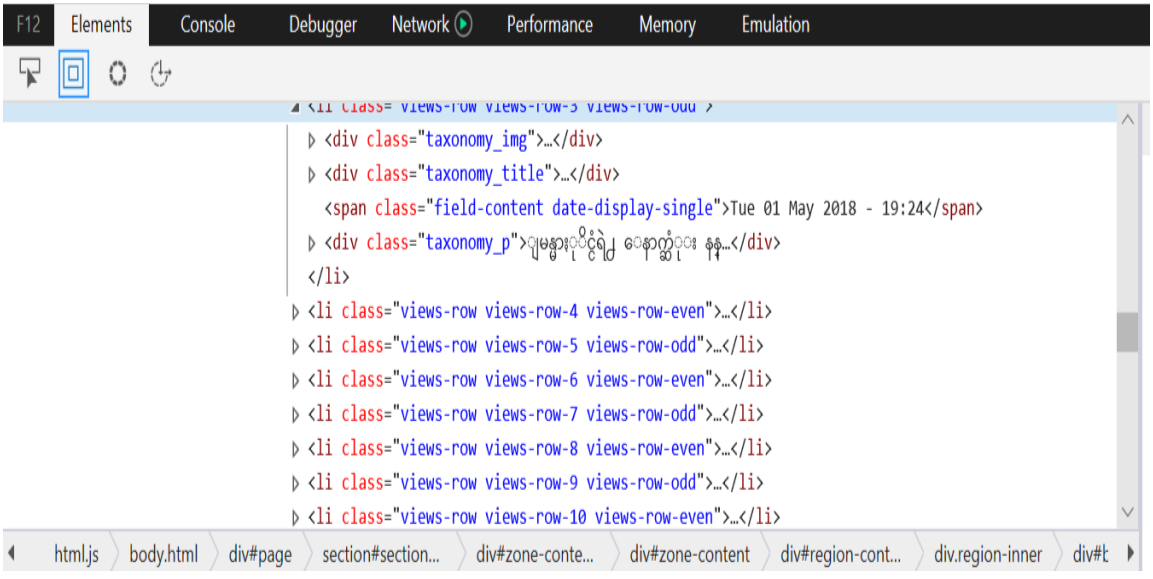


Figure 3.3 How It Looks Like as A Data Entry

3.4 Word Segmentation

Word segmentation is an initial step of processing which is performed in two different ways. The first way, a baseline method, is segmenting the text as an ordinary word and the second one is n-gram-based word segmentation. Then, compares the second way with the baseline method.

The task of word segmentation for languages without word boundaries in writing have been pursued based on the sequence of characters, morphemes (words), syllables and the matching with a prepared dictionary. For the Burmese language, which has no large tagged corpus, the previous work tried to make word segmentation based on a pattern matching approach with a limited size of dictionary, statistical approaches, and syllable-based approaches. However, they have serious problems particularly in coping with unknown words that are not in the training data set and the dictionary. The matching approaches just try to find matches with a series of words in the dictionary and have no way to detect words that are not listed in the dictionary. As a result, unknown words tended to be segmented excessively, as a sequence of multiple short words, in matching. For statistical approaches, more training data is required than other approaches. The tasks that are based on syllable segmentation and syllable merging require lexicon database or dictionary. Thus, their performance depends on the size and quality of the dictionary, though there is no publicly available, fully trustable dictionary for the Burmese language with comprehensive coverage. A large standard corpus of Myanmar Unicode documents is also still missing, let alone

a well-tagged one. Under this situation, a word segmentation method is needed which is not based on the dictionary or a large corpus.

3.4.1 Ordinary Word Segmentation

As a baseline method, the data (text) is segmented as an ordinary word. The Burmese text is a string of characters and includes both single and compound words that consist of more than one single word. A single word consists of one syllable. A syllable is formed based on the rules that are quite definite and unambiguous in Myanmar text. Two or more consonants and characters to be attached to the consonants are combined to form one syllable while vowels are often not explicitly written. Therefore, characters are more basic units than syllables for Burmese text. Generally, spaces are added between phrases to segment the text and the Burmese language has punctuations like comma and sentence delimiter in English, but it does not have word boundaries. In this case, Burmese words are segmented based on the character clusters.

3.4.1.1 Burmese Character Cluster (BCC)

Burmese characters are not ordered at random. Some combinations of characters are possible while other combinations are not possible. The same characteristics is seen in Thai, Lao, and Khmer, and Thai Character Clusters (TCC) has been proposed by Theeramunkong et al., 2000; Tongtep and Theeramunkong, 2010. Based on TCC, I propose Burmese Character Clusters (BCC).

Characters clustering idea is employed in which character clusters mean groups of some inseparable characters due to language characteristics. As described, the Burmese writing system is different but similar to the Thai writing system. Both have four positions (front, back, upper, below) for consonants to add vowel and tone marks, and the Burmese scripts are also separated at three levels (upper, middle, lower). Each cluster pattern in TCC as well as the Burmese scripts are manually checked, and a BCC, as shown in Table 3.1, is proposed. There are 29 cluster patterns, among which 22 of them corresponds to equivalents in TCC and seven of them are newly proposed here (8, 9, 10, 11, 20, 21, and 27 in Table 3.1).

In Table 3.1, ‘vowel’ group includes dependent vowels, independent vowels, medials, and final symbols because the positions of their attachment are the same such as front, back, above and below the consonant. Characters are basic units in the Burmese language. C (Consonant), VF (Front Vowel), VB (Back Vowel), VU (Upper Vowel), VL (Lower Vowel) and T (Tone) are short forms of characters. The meaning of C+VF is a consonant with a front vowel,

C+VB+VU is a consonant, a back vowel and an upper vowel, and C+VF+T means a consonant, a front vowel and a tone marks form one group. In ‘Meaning’ column, all the patterns are clusters of characters.

Table 3.1 Types of Burmese Character Cluster

No.	Types	Meaning	Example	Note
1.	T1	C+VF	မေ အေ ကေ	TCC
2.	T2	C+VB	လာ ချ သာ	TCC
3.	T3	C+VU	သိ ထိ မိ	TCC
4.	T4	C+VL	ကူ လှ မှ	TCC
5.	T5	C+VF+VB	သော ကော ပြော	TCC
6.	T6	C+VF+VU	ငြိ ကြံ့	TCC
7.	T7	C+VB+VU	ချို ချို	TCC
8.	T8	C+VU+VL	မိုနို	Newly Proposed
9.	T9	C+VL+VB	ကွာ ခွာ	Newly Proposed
10.	TF	C+VF+VF	မြေ ခြေ	Newly Proposed
11.	TL	C+VL+VL	လွှမူ	Newly Proposed
12.	T10	C+VF+VB+VU	မော် သော် လော်	TCC
13.	T1T	C+VF+T	မေး အေး ဆေး	TCC
14.	T2T	C+VB+T	လား အား ခါး	TCC
15.	T3T	C+VU+T	သီး ထီး နီး	TCC
16.	T4T	C+VL+T	မူး ဖူး တူး	TCC
17.	T5T	C+VF+VB+T	ဈေး ချေး	TCC
18.	T6T	C+VF+VU+T	ကြီး ငြီး	TCC
19.	T7T	C+VB+VU+T	ကျိုး ဂျိုး	TCC
20.	T8T	C+VU+VL+T	မိုး နိုး	Newly Proposed
21.	T9T	C+VL+VB+T	သွား နွား	Newly Proposed
22.	S	Space		TCC
23.	D	Digit	၁၂၃၄၅၆၇၈၉၀	TCC
24.	C	Consonant	ကခဂဃ...ဪအ	TCC
25.	E	English	A,B,C,….Z	TCC
26.	V	Vowel	ာ ါ ိ ီ ု ူ ေ ဲ ျ ွ ြ ှ ဿ ◌	TCC
27.	A	Abbreviation	၏ ၌ ၍ ၎	Newly Proposed
28.	P	Punctuation	၊ ။	TCC
29.	T	Tone	့ း ◌်	TCC

3.4.1.2 CRF Training

For labeling and tagging the sequence of characters clusters, a CRF model is employed. Most of the previous tasks such as word segmentation, POS tagging, and named entity recognition, employed CRF for labeling the sequence of data and achieved the good performance rate. The largest benefit of CRF is that it can handle the various types of non-independent features of input. The data is manually segmented according to the setup in Table 3.1 and train the CRF segmentation models. 'B,' 'I,' 'E,' and 'S' are used to label the characters' sequence of a word. 'B' means the beginning character cluster of a word, 'I' means the inner or inside character cluster of a word, 'E' means the end character cluster of a word, and 'S' is the single character cluster word. The characters are clustered according to the rules in Table 3.1. For example, the word 'I' for *girl* in the Burmese language is ကျွန်မ. The characters are clustered and labelled as ကျ (B), ွ (I), န် (I) and မ (E). To train a CRF model based on the input data and neighboring labels, a set of feature function needs to be defined. In this case, unigram feature sets are used. After training the CRF model, the constructed model is used to assign the labels for the new data. Based on the predicted labels, the data is segmented. The overview of my CRF model training is illustrated in Figure 3.4.

Although this word segmentation approach did not use the dictionary or tagger, the method still needs supervised answer data. The merit of this segmentation method is that it can give better accuracy than other unsupervised methods. Since there is no pattern matching with the existing lexicon or finding words in the dictionaries, the performance does not depend on these kinds of resources. The main drawback is that the data and the supervising tags to train CRF model need to be prepared manually, which is impossible for large data size. Manual working takes too much time and effort. Therefore, to solve this problem, a new way for word segmentation is proposed and compared with this baseline model.

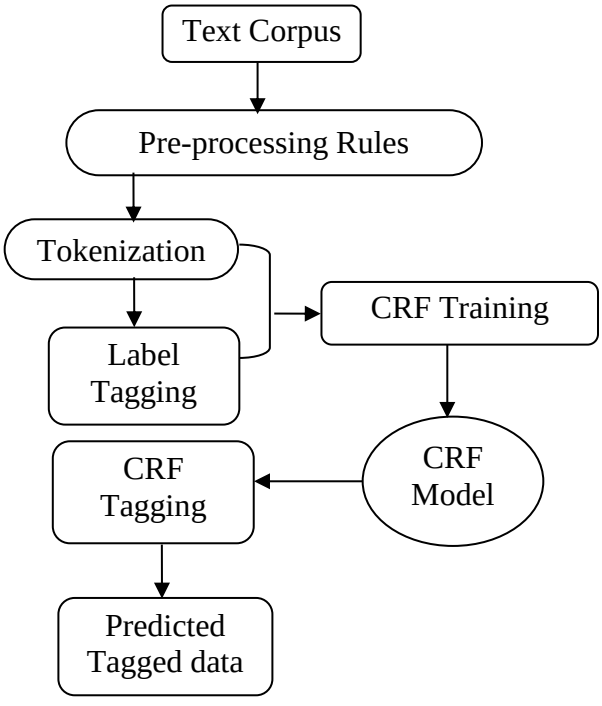


Figure 3.4 Overview of CRF Model

3.4.2 N-gram-based Word Segmentation

As the Burmese language is one of the low resource languages without publicly available, sufficient size of tagged corpora, dictionaries, and other tools, improvements should be expected without using such resources. The method of n-gram-based word segmentation is based on the n-gram characters that do not need to consider the specific language properties without using dedicated tools and resources. The idea of n-gram word segmentation can be applied to all kinds of languages. In addition, unlike ordinary word segmentation, n-gram word segmentation does not need supervised answers.

N-gram-based word segmentation means the data is separated into a sequence based on the number ‘N’ in N-gram. For example, the sentence ‘I am a student’ is separated into ‘Iama’, ‘amas’, ‘mast’, ‘astu’, ‘stud’, ‘tude’, ‘uden’ and ‘dent’ if N is 4 (4-gram). N-gram-based words segmentation method is employed as a newly proposed approach in this work. This is because the proposed approach is focusing on the low resource languages instead of focusing on one specific language and the processing of n-gram word segmentation does not care about the specific characteristics of each language. Among the low resource languages, the Burmese language is used for testing the approach. After segmentation in this work, for the words extraction from all n-gram words, the idea of choosing n is to start from 25 (25-gram) and stop at trigram

because of coping with the construction of the Burmese words. For example, လွှတ်တော်ကိုယ်စားလှယ်များ (congressman), ဒီမိုကရေစီအရေးတော်ပုံကြီး (democracy revolution), စိုက်ပျိုးရေးလုပ်ငန်းများ (agriculture) for 25-gram words and မှာ (at), သည် (subject marker), ငါး (five) for trigram words. From 25 to trigram words segmentations are iteratively processed and extract the frequent words from the whole news articles. The idea of extracting frequent n-gram words is that if some n-gram words frequently occur in the data, it has some meaning even though it is not the same with the meaning for human understanding. The extraction procedure is as follows.

First, 25-gram words segmentation is performed on the pre-processed dataset (the details of pre-processing steps will be explained in Section 5.3), in which all the replaced letters such as ‘X’ for both English and Burmese numbers and ‘W’ for English words are ignored. From all the 25-gram words, extract only the 25-gram words that have the frequency at least five times. Here, five is chosen because most of the n-gram words have very low frequency. The extracted 25-gram words are replaced with the letter ‘P’ in the data. As a next iteration, 24-gram words are segmented on the resulted data set from the previous step (25-gram word segmentation), in this case the replaced letters ‘X’, ‘W’ and ‘P’ are ignored. The words that appear in the data at least 5 times are extracted and are replaced with the letter ‘O’. The same processing is performed in the next iterations, up to trigram (3-gram) words extraction. In each step, 23-gram, 22-gram, 21-gram, 20-gram, 19-gram, 18-gram, 17-gram, 16-gram, 15-gram, 14-gram, 13-gram, 12-gram, 11-gram, 10-gram, 9-gram, 8-gram, 7-gram, 6-gram, 5-gram, 4-gram and trigram extracted words are replaced with ‘N’, ‘M’, ‘L’, ‘K’, ‘J’, ‘I’, ‘H’, ‘G’, ‘F’, ‘E’, ‘D’, ‘C’, ‘B’, ‘A’, ‘9’, ‘8’, ‘7’, ‘6’, ‘5’, ‘4’ and ‘3’ respectively. An example is shown in Figure 3.5.

By doing this way, n-gram words of different sizes are obtained. After segmentation and extraction steps, very high frequency n-gram words (appear more than 500 times) are also removed in the same manner with stopword removal. However, the total number of variable-length n-gram words in a text is still much higher than that of ordinary words in the same text, which will make the vector of each news article more sparse. In order to reduce the number of n-gram words, I will propose the use of distributive word representation models to calculate the similarity among n-gram words for grouping. in the next section.

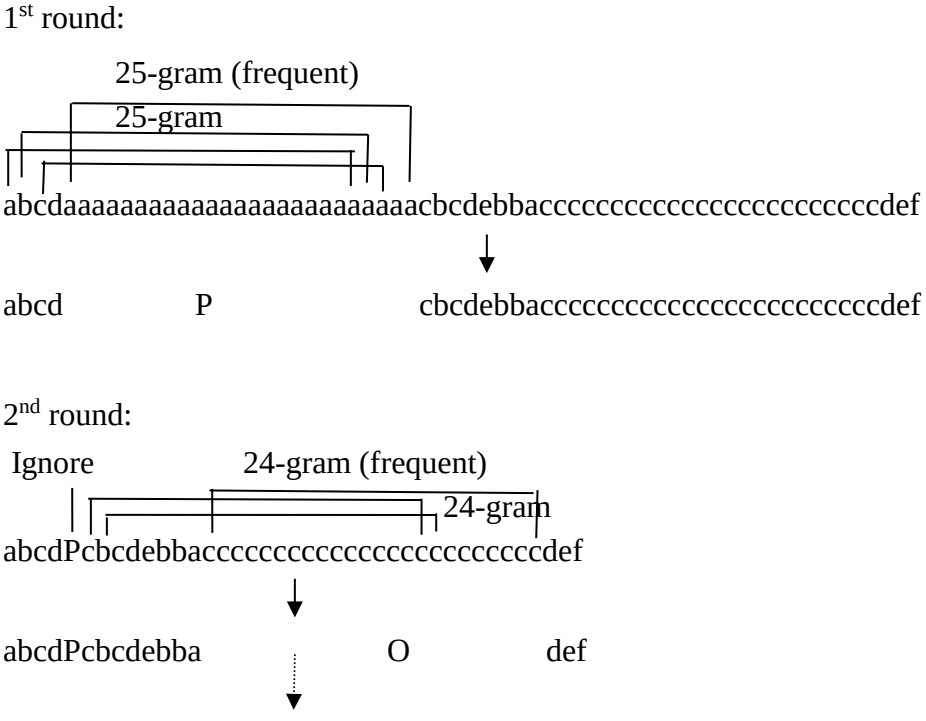


Figure 3.5 Example of Variable-length N-gram Word String Conversion

3.5 N-gram-based Words Grouping by using Distributive Representation of Words

In NLP research areas, different kinds of vector-based representations of words have been popular and widely used by many researchers because of their great performance. In this research, two popular distributive words representation models are applied, namely, word2vec and GloVe. Both word2vec and GloVe work in a very similar way. They want to build a vector space where the word vectors are learned based on the occurrence of words in the data locally or globally. Therefore, the position of each word is influenced by its neighboring words based on their context and semantics. Word2Vec learns local individual of word co-occurrence pairs and GloVe model starts with global co-occurrence statistics across all words in the corpus.

They are unsupervised methods. The statistics, the frequencies of word appearance, in a corpus is the primary source of information available for all unsupervised methods to learn word representations. The underlying assumption is that the words in similar contexts have similar meanings. Intuitively, the words that share many contexts will be similar to each other. Similarly, the contexts sharing the same set of many words will also be similar to each other. GloVe and

word2vec are considered to be able to encode at least a fragment of meaning based on contexts. In other words, the similarity of word vectors reflects at least a fragment of meaning similarities.

The reason for applying these models in this research is that some n-gram words that have similar meanings to be grouped. In the previous section, n-gram-based word segmentation is already explained. The number of n-gram words is usually much bigger than the number of ordinary words with the same data size. Therefore, after segmentation and extraction, the words that have similarity are grouped in order to reduce the number of n-gram words. For the words grouping in both models, cosine similarity is used to group n-gram words. In this case, words grouping is based on different cosine similarity degree, 70.0%, 80.0% and 90.0%, and compares the results. As I mentioned about cosine similarity in Section 2.6.2.3, the similarity is calculated based on the following equation:

$$cos\ \theta = \frac{\vec{a}\ .\ \vec{b}}{\|\vec{a}\|\ \|\vec{b}\|} \tag{3.1}$$

3.6 Sentiment Values Calculation and Sentiment Classification

Many researchers have proposed many ways to calculate the sentiment values of the words which are stored in the sentiment dictionary. Among them, this study adopts the method introduced by Turney and Littman (2003). Their method calculates the semantic orientation, the evaluative character of a word, for each word from its associated statistical information with a set of positive and negative seed words. They did not use any available dictionaries to determine the sentiment value of a word and the method gave a good result, above 95.0% accuracy of classifying positive and negative words. In this study, existing dictionaries or lexicon are not available and the goal is to cope with the problem of lack of resources problems.

Turney and Littman (2003) method includes two main steps. As the first step, Latent Semantic Analysis (LSA) is calculated. Calculation of LSA includes two main parts. First tf-idf (term frequency – inverse document frequency) matrix, the original matrix X, is constructed for the pre-processed data set. In the matrix, the row vectors represent the words, and the column vectors represent the documents (news articles). Each cell represents the weight, tf-idf score, of the corresponding word in the corresponding document. The next step is to apply Singular Value Decomposition (SVD) to compress the original matrix, X. SVD decomposes the matrix X into a product of three matrices, $X = UDV^T$, where U and V are orthogonal matrices and D is a diagonal matrix of singular values. Let D_k, the diagonal matrix with the top k singular values, U_k and V_k formed by selecting the corresponding columns from U and V. The matrix $U_k D_k V_k^T$ can be

considered as a compressed version of the original matrix. LSA works by measuring the similarity of words vectors in the compressed matrix instead of using the original matrix. The similarity of two words, LSA (word1, word2), is measured by calculating the cosine similarity of two corresponding vectors in the compressed matrix.

The second step is calculating SO-LSA. Semantic Orientation, SO-LSA of a word is calculated based on the following equation:

$$SO\text{-}LSA(word) = \sum_{pword \in Pwords} LSA\,(word, pword) - \sum_{nword \in Nwords} LSA\,(word, nword) \quad (3.2)$$

Here, Pwords means the set of positive seed words and Nwords means the set of negative seed words. For this calculation, 8 positive seed words and 8 negative seed words are selected. The same number of seed words for positive and negative are chosen because the distribution of positive class and negative class is similar in the data. These seed words are originally chosen words that they appear frequently when manually inspecting the news articles in the data.

If SO-LSA (word) value is positive, the word is classified as a positive word and it is a negative word when SO-LSA (word) value is negative. The values for all the words from all the 500 news articles are calculated and the sentiment value of each word are stored in the sentiment dictionary. For classification, only 437 news (226 positive news and 211 negative news) are used for experiments because the rest are neutral news. To classify the news, the sentiment values, SO-LSA, of all the words in each news are added. If the resulting value is positive, the news is positive. Otherwise, it is negative.

3.7 Two Types of Sentiment Classification

In this study, I employ two types of classification for the news articles.

3.7.1 Summation of Sentiment Values

The classification of each news article is based on the summation value of all words in that article. If the resulted value is positive, the article is classified as 'positive.' Otherwise, it is classified as 'negative.' The formula is as follows:

$$\sum_{i=1}^{i=n} V_{wi} \quad (3.3)$$

where w_i is the i-th word in a news article and V_{wi} is the sentiment value of w_i.

3.7.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the most popular and efficient machine learning algorithms. The objective of SVM is to find the optimal separating hyperplane which separates the training data and maximizes the margin of the training data. There are several separating hyperplanes which separate the data set successfully. SVM finds the optimal separating hyperplane which is located as far as possible from each category of data points. This kind of hyperplane provides the best performance for classification. According to the goal of SVM, it needs training data. Therefore, this is a supervised machine learning algorithm.

In this work, SVM is employed with a linear kernel using scikit-learn with default parameter setting except for *C* value which is 0.1 or 1.0. I made a pre-processing making the data into a bag-of-words format, ignoring the word order in the data. We set the features of each word/n-gram word as its tf-idf value and sentiment value for both experiments.

3.8 Concluding Remarks

In this chapter, the main steps of research methodology are explained. The construction of distributed word vectors for many words in a target language requires a large amount of text data that are already segmented by words. However, particularly for many low resource languages, the preparation of that sort of big text data is difficult. Thus, there have been only a small number of attempts to apply distributive word models to such low resource languages. There has been no attempt to apply a distributed word representation model to n-gram words. The evaluation of CRF-based Burmese word segmentation and details of data description, experimental conditions will be discussed in the next chapter.

CHAPTER 4

EVALUATION OF CRF BURMESE WORD SEGMENTATION

4.1 Introduction

As the first part of research methodology that is mentioned in 3.4.1, I propose a CRF-based word segmentation method with Burmese Character Clusters (BCC) for the Burmese language. In this chapter, experimental evaluation is conducted for this method.

4.2 Experimental Setting

For this experiment, three news articles are picked up to test the experiment which consists of 7,035 words. The size of this data is quite small, due to the high cost of preparing such data, but this also aims to evaluate how well a word segmentation method works with a tiny size of data, considering the hard availability of such data in low resource languages. After grouping some single words into compound words, the data contains 3,003 words in a total of 167 sentences. This data is divided into five groups for 5-fold cross-validation, using 4 groups for a training set and the remaining group for a test set with five iterations. The data and the supervising tags are manually prepared by the author. For labelling, 'B,' 'I,' 'E,' and 'S' labels are used and for feature set, unigram feature set $\{x_{t-1}, x_t, x_{t+1}\}$ is used to train the model.

In this work, three approaches are compared: word segmentation with BCC, without BCC and syllable-based word segmentation. The answer is manually tagged and predict the result in a machine learning framework. For all evaluations, CRF++ toolkit, an open source implementation of CRF, is employed to build a CRF model for manually segmented data (Kudo, 2013).

4.3 Result and Discussion

The performance including accuracy, precision, recall, and F-measure of segmentation with BCC, without BCC and syllable-based are compared in Table 4.1, Table 4.2 and Table 4.3 respectively.

Table 4.1 Performance of with BCC Approach

Iteration	Accuracy	Precision	Recall	F-measure
#1	0.980	0.992	0.981	0.986
#2	0.982	0.992	0.985	0.988
#3	0.991	0.995	0.998	0.992
#4	0.993	0.992	**0.998**	**0.994**
#5	**0.994**	**0.996**	0.988	0.992
Average	0.988	0.992	0.990	0.991

Table 4.2 Performance without BCC Approach

Iteration	Accuracy	Precision	Recall	F-measure
#1	0.874	0.930	0.980	0.954
#2	0.896	0.948	0.975	0.961
#3	**0.954**	**0.953**	**0.994**	**0.973**
#4	0.937	0.922	0.990	0.954
#5	0.914	0.904	0.986	0.943
Average	0.915	0.931	0.985	0.957

Table 4.3 Performance of Syllable-Based Approach

Iteration	Accuracy	Precision	Recall	F-measure
#1	**0.860**	**0.896**	0.786	0.833
#2	0.842	0.893	**0.812**	**0.850**
#3	0.800	0.684	0.810	0.742
#4	0.810	0.766	0.654	0.705
#5	0.808	0.702	0.785	0.745
Average	0.824	0.788	0.769	0.778

Table 4.4 Average Performance of Each Approach

	With_BCC	**Without_BCC**	**Syllable_based**
Accuracy	**0.988**	0.915	0.824
Precision	**0.992**	0.931	0.788
Recall	**0.990**	0.985	0.769
F-measure	**0.991**	0.957	0.778

Table 4.4 presents the average performance of each approach. By comparing the results, word segmentation with BCC achieves the best performance among the three approaches in this study. The performance of segmentation with BCC is significantly higher than that of segmentation without BCC and obviously higher than that of syllable-based segmentation, where each accuracy is 0.988, 0.915 and 0.824 respectively. Similarly, for the F-measure, BCC produces better results than the other two ways: 0.991, 0.957 and 0.778.

The input features for the process of word segmentation with BCC is important because the ways of characters clustering are based on the structure of the Burmese words. So, the rules are common in most of the words form and this process gives the best performance. It still makes a few wrong segmentations after a word with a tone mark. The processing without BCC has lower accuracy and F-measure than processing with BCC. This result comes from the fact that there are much more spelling variations without BCC. Most of the errors appear after the words with a final symbol or a tone mark. When a word appears twice sequentially for some reason, there was an occasional error for the second word segmentation. Syllable-based processing produces more errors than the other two ways. Most of the errors occurred after a final symbol or a tone mark and also occurred in some sequences of consonants. Most of the words that have a final symbol, or a tone mark appear at the end when the segmentation process is made. Therefore, there are errors in some middle words. The Burmese consonants can be standalone to form a word. The standalone consonants can be combined with other words to make a complete meaning or a single word. In such cases, the process could not recognize the consonants when they appear as a single word. The performance of syllable-based segmentation might be higher if the various n-gram were used (bi-gram, tri-gram, 4-gram, or 5-gram); however, this study tried only unigram to keep the condition same in all the experiments.

As stated, these three ways produced some errors particularly after tone marks because the Burmese words which have tone marks mostly appear at the end of the sentence or are the final words when word segmentation is made. For example, the word “တိုးတက်” which means

“improve” includes two words “တိုး” and “တက်”, and three consonants “တ”, “တ”, and “က”. These two words must be combined to have the meaning of “improve”. There is a different meaning if these words are separated because each word has their own meaning and after combining these two words, it gives another meaning. In other words, such errors are related to another higher-level task of ambiguity resolution, which is also a future task. The segmentation error appeared at the first word where the tone mark is “း”, the consonant is “တ”, and “ိ” and “ု” are vowels. The process separates the combined word into two words when it meets the tone mark. This leads to a different meaning.

For the data set, training and testing data, words are manually segmented, and no dictionaries are applied in all evaluations. So, the performance of this work does not depend on the quality of the dictionary to be employed. Moreover, there are no skipping characters, syllables or words during processing since the pattern or word matching with the existing corpus or lexicon database is also not used in this work.

As the data employed are of a small size, the number of unknown words, which appears only in the test set of each iteration, is also small, but the result still shows that such unknown words are also mostly segmented correctly. For example, the word “အတွက်” can be used as “for”. This consists of two words “အ” and “တွက်”, three consonants “အ”, “တ”, and “က”. The symbol “်” is called the final symbol, which appears mostly at the end. Therefore, the word is segmented correctly although it is not contained in the training set.

4.4 Concluding Remarks

The experimental conditions for the other steps of research methodology will be explained in the next chapter.

CHAPTER 5

EXPERIMENT CONDITION FOR SENTIMENT ANALYSIS

5.1 Introduction

This section explains the details of all experiments in this research. In the next section, the employed data are explained. In 4.2.1, the questionnaire survey and the sentiment value assignment for each news article based on the survey are described. Then, four kinds of experiments for the Burmese news articles data and three experiments for the Amazon Product Review data are processed in comparative ways.

5.2 Data

5.2.1 Burmese News Articles Data

In this work, two types of data are employed. The first data is newspaper articles of *7Day Daily*, one of the most popular private newspapers in Myanmar and the second one is Amazon Product Review data. The first data, daily news articles, can be obtained from its official website (*7Day Daily*)[11]. On the website, the news articles are grouped according to the categories: Election, News, Business, Life, Entertainment, Opinion, Politics, Regional, Sports, World News, Cartoon, Videos, and Special Reports. So, the website provides the information by categorizing the news in daily newspapers. Among them, the news articles only from the Opinion section are collected because the Opinion category contains the opinions from many people about the news. Thus, they can be applied for sentiment classification.

The reasons why I chose this newspaper are as follow. There are two main types of newspapers in Myanmar. They are government-owned newspapers, namely Myanmar Alin, Kyemon, The New Light of Myanmar, Yadanarpon and Mandalay, and private newspapers including *7Day Daily*, *Eleven*, *The Voice* and so on. Among them, *Myanmar Alin*, *Kyemon* and *The*

[11] Cited from: http://www.7daydaily.com/

New Light of Myanmar are daily newspapers based in Yangon, Myanmar and distributed to most of the states and divisions of Myanmar. Yadanarpon and Mandalay newspapers are also daily newspapers that carry mainly Mandalay and Upper Myanmar-related news. Some private newspapers are issued on a daily basis, and some are distributed weekly around the country. *7Day Daily* is a daily one, distributed around the country.

First, I start to collect the data in April 2017. In May 2017, I collected 500 news articles and the period of these news articles is from December 2016 to May 2017. I made the surveys for 500 news articles. For data processing, I collected more news articles in each month of 2017. At the end of December 2017, I collected another 780 news articles. Finally, the total number of articles is 1,280 and the period of all news articles is from December 2016 to December 2017.

5.2.2 Questionnaire Surveys and Sentiment Assignment to News Articles

I made surveys for 500 news articles in May 2017 to obtain the sentiment judgments from the Burmese people whether the news is positive or negative. The request form for questionnaire surveys is shown in Figure 5.1.

I distributed the surveys for 500 news articles to 100 university students in Mandalay, Myanmar. I obtained the results from all 100 students at the early of July. From the results of the surveys, “Positive” and “A Little Bit Positive” news articles were considered positive. “A Little Bit Negative” and “Negative” were considered as negative. Two persons were assigned for each news article. When the polarity was the same, the polarity was adopted. When the polarity differed between two persons, another person was asked to rate it, and the majority decision was adopted. Among 500, 154 articles had different results, and another 30 students were asked to rate the articles. Therefore, every article was read by at least two people. Then, the final decision for each article was made based on the majority, and it was done in August 2017. The age of all participants was from 20 to 24, and the proportion of male and female was 34% and 66% respectively. Among 500 news, the number of positive news was 226, the negative news was 211, and neutral news was 63.

The survey data of 500 news are used for calculating the sentiment values of each word, and only 437 news (positive and negative news among 500 news) is used for classification. The performance is evaluated based on the classification result.

Dear Participant,

My name is Myat Lay Phyu and I am a graduate student at Prince of Songkla University. As part of my Master thesis, I am conducting a survey that investigates the conditions on whether news articles contain positive, neutral, or negative contents. I would like to invite you to participate in this research study by completing the attached survey. According to your opinion about the news, please choose one of the choices "Positive", "A Little Bit Positive", "Neutral", "A Little Bit Negative", and "Negative". I will appreciate if you could complete this survey. Thank you for your kind cooperation and taking the time to assist me in my educational endeavour. Your responses will contribute to this academic research.

Sincerely,

Myat Lay Phyu

Figure 5.1 Survey Request Form

5.2.3 Amazon Product Review Data

The second data, Amazon Product Review data, can be obtained from He, R., and McAuley, 2016; McAuley et al., 2015. Each review contains id, ProductId, UserId, ProfileName, HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary, and Text. Among these kinds of information, I use only Score and Text. The score is from 1 (very negative) to 5 (very positive), and the text is review comment. The reviews which have the score of 1 or 2 are considered as negative reviews, and 4 or 5 are positive reviews. The purpose of using this dataset is for a bigger dataset similar to Burmese with supervised labels, which is not available in Burmese. For the Burmese news articles, I have to make surveys to get the classification results which takes time to collect the data and results. Therefore, I use this data as a pseudo-Burmese dataset to evaluate the proposed method with large data size. Some pre-processing steps are performed for the Amazon Product Review text in order to have similar characteristics with the Burmese language which will be explained in Section 5.4. 10,000 positive reviews and 10,000 negative reviews are collected. With this data, only the proposed approach is tested.

5.3 Experiment Setting for Burmese News Articles Data

Four ways to perform sentiment analysis of the Burmese news articles are processed in comparison. As a first step, the same data pre-processing is performed. After that, the data is segmented in comparative ways. The segmented data is used to calculate the sentiment

values of each word and construct sentiment dictionary. In this step, the same methodology is applied for calculating sentiment values and constructing sentiment dictionary. Finally, sentiment classification is performed in two ways. The first way is the summation of sentiment values of all words that are contained in each news article. If the summation value is positive, the news is positive. Otherwise, the news is negative. The second one is classification with supervised machine learning technique: Support Vector Machine (SVM).

Table 5.1 Preprocessing part of Experiments

Experiment I	Experiment II
1. Use all news articles (1,280 news)	1. Concatenate all news articles (1,280 news) and insert '®' at the end of each news
2.Replace English punctuations !"#$%&'()*+,-./:;<=>?@[\]^_`{|} ~ with a 'space'	2. Replace English punctuations !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ with 'U'
3. Ordinary Word Segmentation with CRF	
4. Remove English words and Numbers (both Burmese and English numbers)	3. Replace English words with 'W' 4. Replace numbers (both Burmese and English numbers) with 'X'
5. Replace Burmese punctuation: a comma with a 'space'	5. Replace Burmese punctuations: comma and full stop with '$'
6. Replace Burmese punctuation: a full stop with a '$'	
	6. N-gram-based word segmentation
7. Remove stop words	7. Remove n-gram words which have a high frequency

5.3.1 Experiment I

In this experiment, the dataset contains 1,280 news articles which contain 1,008,274 ordinary words. The data is segmented by using Burmese characters clustering with a machine learning framework, a Conditional Random Field (CRF) model, as explained in Section 3.4.1. The obtained CRF model is employed to annotate a new set of unsegmented data. The automatic annotation of this data is manually corrected. From the segmented data, the words that are important to classify the opinions and appear frequently are selected as positive and negative

seed words. The stop words list is used to remove some unimportant words from the data. The stop words list is manually prepared by inspecting some few hundreds of articles because there is no publicly available stop words list for the Burmese language. The list contains 278 words. After removing stop words, the frequency of each word is counted in the whole dataset and extract the words that appear at least five times in the dataset. The extracted data from 1,280 news contains 642,887 words. Among 1,280 news, 500 news which already has classification results contain 115,144 words from which the number of unique words is 8,980. The chosen seed words are as follows:

Positive: ['ကောင်းသော', 'ဖွံ့ဖြိုး', 'တိုးတက်ရေး', 'ပျော်ရွှင်သော', 'အောင်မြင်', 'အဆင်ပြေ', 'မှန်ကန်သော', 'ငြိမ်းချမ်းရေး']

Negative: ['ကျဆင်း', 'အဆင်မပြေ', 'ပြဿနာ', 'ဆိုးသော', 'အခက်အခဲ', 'အန္တရာယ်', 'ဝေဖန်မှု', 'ဒုက္ခ']

The translation of these seed words are:

Positive: [‘good’, ‘develop’, ‘improvement’, ‘happy’, ‘success’, ‘convenient’, ‘correct’, ‘peace’]

Negative: [‘decline’, ‘inconvenient’, ‘problem’, ‘bad’, ‘difficulty’, ‘danger’, ‘criticism’, ‘trouble’]

These seed words are used for the next step, calculating the sentiment values and constructing sentiment dictionary. For this step, LSA and SO-LSA methods are performed to calculate the sentiment values and construct sentiment dictionary. The classification for each news is performed based on the summation of sentiment values of all words in each article.

5.3.2 Experiment II

This experiment uses the same dataset as the previous experiment, 1,280 news articles. During variable-length n-gram words segmentation, n-gram words of a very low frequency (lower than five times) are removed. After segmentation process, very high-frequency words (more than 500 times) are removed like removing stop words in ordinary word segmentation case. In this case, the frequency numbers are chosen after manually inspecting the data. There are many kinds of n-gram words which rarely appear and a few n-gram words that occur more than 500 times in the data. After that, 1,280 news data contain 894,502 n-gram words. Among 1,280 news, 500 news which already have classification results contain 223,440 n-gram words from which the number of unique n-gram words is 78,648. The same seed words list with the previous experiment is applied in this case.

Note that an n-gram word containing a seed word but not containing a negative word is also regarded as the same seed word. After selecting the seed words, the same methodology, LSA and SO-LSA methods are processed and classify the news articles.

5.3.3 Experiment III

All the extracted n-gram words of 1,280 news articles from the previous Experiment II are used in this case. The extracted data contains too much variant of n-gram words, and the data size is bigger than the original data size. Therefore, as the next step in this experiment, distributed word representations model is employed to group some n-gram words that have similarity to reduce the data size. In this experiment, word2vec model is trained where Skip-gram model is employed, Hierarchical softmax for training Skip-gram model, minimum occurrence = 5, a symmetric context window of size 10, 300-dimensional vectors and the words grouping is based on cosine similarity of 70.0%, 80.0%, and 90.0%. After grouping the words, among 1,280 news, 500 news which already has classification results contain 223,437 n-gram words from which the number of unique n-gram words for each similarity is 43,360, 75,710 and 78,287 respectively. Then, the same procedure, the processing of LSA and SO-LSA by using the same seed words list and the classification, is performed.

5.3.4 Experiment IV

This experiment also uses all the extracted n-gram words of 1,280 news articles from Experiment II. However, in this case, another distributed word representation model called GloVe is applied for words grouping. For parameter setting, maximum vocabulary size = 100,000, scaling factor = ¾, learning rate = 0.05, minimum occurrence = 5 and use a symmetric context window of size 10. The model is run 100 iterations for vectors 300 dimensions and the words grouping is based on cosine similarity of 70.0%, 80.0% and 90.0%. After grouping the words500 news, among 1,280 news, which already have classification results contain 223,437 n-gram words from which the number of unique n-gram words for each similarity is 78,644, 78,645 and 78,648 respectively. Then, the same procedure, the processing of LSA and SO-LSA by using the same seed words list and the classification, is performed.

5.4 Experiment Setting for Amazon Product Review Data

For second data, 20,000 Amazon Product Review data, the proposed approach: a character-based variable-length n-gram word model with distributive word representation techniques and without words grouping are tested. With this data, only three experiments: n-gram-based words without word groupings, n-gram-based words with word2vec word groupings and n-gram-based words with GloVe word groupings are performed. The data contains 1,168,017 ordinary words. As the aim of using the Amazon Product Review dataset is to prepare a larger dataset similar to the Burmese language with supervised answers, some pre-processing steps are necessary. The Burmese language has no word segmentation with spaces and no inflected forms. Thus, as a part of pre-processing, all the inflected forms are changed into their base form using the Stanford POS tagger and all the spaces are removed.

After that, the same data processing steps with the processing of Burmese news articles are performed except in the case of deciding 'n' in n-gram-based segmentation. For Burmese data processing, the segmentation step starts from n=25 (25-gram), but for the Amazon Product Review data, it starts from 10 (10-gram). Since the reviews are written in English, 10-gram can cope with ordinary words. For SO-LSA calculating step, the following seed words which frequently appear in the data are chosen.

Positive: ['good', 'great', 'love', 'delicious', 'favorite', 'easy', 'excellent', 'perfect']
Negative: ['bad', 'disappoint', 'problem', 'waste', 'awful', 'horrible', 'weak', 'nasty']

These seed words are used in all three experiments. Note that an n-gram word containing a seed word but not containing a negative word is also regarded as the same seed word. The parameter settings for word2vec and GloVe training and choice of cosine similarity are also same with the processing of Burmese news data.

5.5 Computation Environment

For all the experiments, I use Python programming language (Python version 3.6) with NumPy and scikit-learn libraries. The CPU of the machine is Intel Core i7-4720HQ with the speed up to 3.6 GHz and memory is 12 GB.

5.6 Concluding Remarks

In summary, there are two types of data for experiments: Burmese news articles and Amazon Product Review. For first data type, four different kinds of experiments are performed. Experiment I in which data is segmented as ordinary words by using CRF-based machine learning framework. For Experiment II, the data is segmented as variable length n-gram-based words without using distributive word representation models. Experiment III and IV have the same data segmentation with Experiment II. However, they use distributive word representation models for words grouping. Experiment III uses word2vec model while Experiment IV employs GloVe model.

For the second data type, three types of experiments are compared. For Experiment I, the data is segmented as variable length n-gram-based words without using distributive word representation models. Experiment II and III have the same data segmentation with Experiment I. However, they use distributed word representation models for words grouping. Experiment II uses word2vec model while Experiment III employs GloVe model. Therefore, the experiment conditions of Experiment I, II and III of Amazon Product Review are the same with Experiment II, III and IV of Burmese news articles respectively.

The details of experimental conditions are explained in step by step. The results and discussion of all experiments will be explained in the next chapter.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Introduction

In the previous chapter, the processing steps of all experiments were explained. In this chapter, the extracted data in each experiment, experimental results and discussion will be explained.

6.2 Burmese News Articles

Among 1,280 news articles, the number of words in 500 news articles is described in Table 6.1. These data were also used to calculate the sentiment values.

Table 6.1 Data Description for Burmese News Data

Number of extracted words in 500 articles	Usual words with CRF (Experiment I)	N-gram words (Experiment II)	N-gram words after word2vec (Experiment III)	N-gram words after GloVe (Experiment IV)
Total Words	115,144	223,437	223,437	223,437
Unique words	8,980	78,648	43,360 (70%) 75,710 (80%) 78,287 (90%)	78,644(70%) 78,645(80%) 78,648(90%)

6.2.1 Result

The results of two different classification methods for the Burmese news data are shown in Table 6.2 and Table 6.3. Using simple sentiment values calculation in Table 6.2 gives around 74.0% accuracy and 75.0% F-measure as the best result while tf-idf values with SVM

provide 76.0% accuracy and 78.0% F-measure and sentiment values with SVM obtains around 85.0% in both accuracy and F-measure as the best performance.

The results of simple sentiment value calculation for each experiment are shown in Table 6.2. Overall, the performance of n-gram-based word segmentation process is similar to usual word segmentation even though n-gram-based word approach contains too much data. Experiment III and IV show three kinds of results with various cosine similarity of 70.0%, 80.0%, and 90.0% respectively. With the word grouping with word2vec, the performance is a little bit decrease in 80.0% and 90.0% cosine similarity while it is significantly decreased in 70.0% similarity. The result of 90.0% similarity in GloVe (Experiment IV) is the same as Experiment II.

According to F-measure, Experiment I achieved the highest result while GloVe-based word groupings with cosine similarity of 80.0% achieved as high as that. According to accuracy, variable-length n-gram word and GloVe-based word groupings with the similarity of 80.0% produce the best result, and the ordinary word segmentation is a little bit lower. In both word2vec and GloVe word groupings, 80% similarity give better accuracy than 70.0% and 90.0% similarity.

The results of classification with SVM for each experiment are shown in Table 6.3. In both accuracy and F-measure, word2vec word grouping of 80.0% similarity achieved the best result. Like in Table 6.2, the performance of GloVe word grouping is similar to n-gram-based words without word grouping because GloVe made only a few words grouping and most of the words in the text were coped with as they were. In comparison with Table 6.2, Table 6.3 can give the better performance. Overall, in Table 6.3, the performance of n-gram-based words with or without word groupings is a little bit better than that of usual words segmentation.

Table 6.2 Classification with Sentiment Values Summation

	Accuracy	**Precision**	**Recall**	**F-measure**
Experiment I (Ordinary words)	73.23%	72.80%	76.99%	**74.84%**
Experiment II (n-gram words)	72.08%	**76.00%**	67.26%	71.36%
Experiment III (word2vec - 70%)	59.73%	56.44%	**96.90%**	71.34%
Experiment III (word2vec - 80%)	69.34%	70.54%	69.91%	70.22%
Experiment III (word2vec - 90%)	65.22%	65.55%	69.03%	67.24%
Experiment IV (GloVe - 70%)	67.96%	64.53%	84.51%	73.18%
Experiment IV (GloVe - 80%)	**73.68%**	75.34%	73.01%	74.16%
Experiment IV (GloVe - 90%)	72.08%	**76.00%**	67.26%	71.36%

Table 6.3 Classification with SVM

	SVM	Accuracy	Precision	Recall	F-measure
Experiment I (Ordinary words)	Tf-idf values	74.59%	**76.11%**	74.44%	75.27%
	Sentiment Values	83.04%	84.35%	**83.72%**	84.04%
Experiment II (n-gram words)	Tf-idf values	**75.72%**	74.57%	82.16%	**78.18%**
	Sentiment Values	84.22%	87.50%	81.76%	84.53%
Experiment III (word2vec - 70%)	Tf-idf values	73.41%	71.64%	82.66%	76.76%
	Sentiment Values	82.61%	84.52%	81.84%	83.16%
Experiment III (word2vec - 80%)	Tf-idf values	75.48%	74.42%	81.66%	77.87%
	Sentiment Values	**84.68%**	**89.07%**	80.54%	**84.59%**
Experiment III (word2vec - 90%)	Tf-idf values	73.86%	70.99%	**86.52%**	77.99%
	Sentiment Values	83.54%	87.20%	80.96%	83.96%
Experiment IV (GloVe - 70%)	Tf-idf values	75.71%	74.57%	82.16%	**78.18%**
	Sentiment Values	82.84%	86.02%	80.27%	83.05%
Experiment IV (GloVe - 80%)	Tf-idf values	75.71%	74.57%	82.16%	**78.18%**
	Sentiment Values	81.96%	85.02%	80.03%	82.45%
Experiment IV (GloVe - 90%)	Tf-idf values	**75.72%**	74.57%	82.16%	**78.18%**
	Sentiment Values	84.22%	87.50%	81.76%	84.53%

6.2.2 Discussion

In the case of actual word segmentation (Experiment I), stop words and the words that appear lower than five times are removed. For n-gram-based word segmentation (Experiment II), among the n-gram words from the original data, the words that appear at least five times are extracted in each segmentation step. There are various kinds of n-gram words that occur lower than five times in the data. So, I choose at least five times to extract the data, and the same decision is made in Experiment I. After processing segmentation with all n-gram (from 25-gram to 3-gram), n-gram words that appear more than 500 times are removed. After manually checking the data, there are a few n-gram words that occur more than 500 times in the data. Therefore, they are removed like stop words removal in usual word segmentation case. However, the number of extracted n-gram words is still larger than that of usual words for the same data size.

Therefore, I tested other ways (Experiment III and IV) to reduce the number of n-gram words. In Experiment III and IV, 70.0%, 80.0%, and 90.0% represent cosine similarity. With

the word grouping with word2vec (Experiment III), around 45.0%, 4.0% and 0.5% of unique n-gram words are grouped in cosine similarity of 70.0%, 80.0%, and 90.0% respectively. With GloVe (Experiment IV), less than 0.01% of unique n-gram words are grouped in 70.0% and 80.0% cosine similarity. 90.0% similarity degree cannot group any words. This means that word2vec word grouping is more efficient than GloVe word grouping for this dataset. As can be seen in Table 6.1, the resulted number of words in each experiment are the extracted data after segmentation and extraction steps.

The performances of Experiment I and II are similar in Table 6.2. In both cases, only frequent words are extracted and used for processing. It can be considered that frequent n-gram words have some special meaning like actual words for processing, although they may not be meaningful for human understanding. In Experiment III and IV, cosine similarity values are chosen for comparison. In the case of choosing similarity, if a lower cosine similarity is chosen, the more n-gram words will be grouped, and the more data is reduced. However, it can also group the words that have different meanings because their cosine similarity is low. Otherwise, if higher cosine similarity is selected, a few words are grouped, and the data still has different kinds of features which can have bad effects on processing.

After the word grouping with word2vec, the performance is a little bit decrease in 80.0% and 90.0% cosine similarity while it is significantly decreased in 70.0% similarity. This may be too much word grouping, and words with very different meanings are grouped together excessively. Since the similarity degree is low, many different words are grouped as the similar meaning of words. In the case of GloVe (Experiment IV), the result of 90.0% similarity is the same as Experiment II. This is because 90.0% cosine similarity cannot group any words, means that there are no n-gram words which have cosine similarity of at least 90.0%. Since GloVe model can group only a few words in this data, the performance is similar to n-gram-based words without word groupings.

Classification with SVM can give the better performance than simple sentiment values calculation. For SVM classifiers, training and testing set data are based on 10-fold cross validation in all experiments. The reason for dividing the original data into training and test datasets is to use the test dataset as a way to estimate how the model is trained well on the training dataset. The model classifier predicts the result of test dataset, new previously unseen data, which had not been seeing during training but has the same attributes as the training dataset. Basically, cross validation gives more stable and reliable estimates of how the classifiers likely to perform by running multiple different train and test splits and then averaging the results instead of relying entirely on a single particular training set. Mostly, 10-fold cross validation is applied to split the

data. The performance of classification by using sentiment values is better than using tf-idf values in all experiments. This may be because of the small size of data in this experiment. Using tf-idf features can give the better performance if the data is sufficiently enough.

6.3 Amazon Review Data as Pseudo-Burmese

The number of words in 20,000 Amazon reviews data is shown in Table 6.4. It shows the original dataset and extracted dataset in each experiment. This data is used to calculate the sentiment values.

Table 6.4 Data Description for Amazon Reviews Data

Number of extracted words in 20,000 reviews	**N-gram words (Experiment II)**	**N-gram words after word2vec (Experiment III)**	**N-gram words after GloVe (Experiment IV)**
Total Words	658,170	658,170	658,170
Unique words	63,955	25,022 (70%) 33,791 (80%) 62,561 (90%)	63,897 (70%) 63,931 (80%) 63,953 (90%)

6.3.1 Result

The results of two different classification methods for the Amazon reviews data are shown in Table 6.5 and Table 6.6. Using simple sentiment values calculation in Table 6.5 gives around 68.0% accuracy and 72.0% F-measure as the best result while tf-idf values with SVM provide 90.0% accuracy and F-measure and sentiment values with SVM obtain around 86.0% in both accuracy and F-measure as the best performance.

The results of simple sentiment value calculation for each experiment are shown in Table 6.5. In this case, word2vec word groupings give the better performance than n-gram-based words without word grouping and GloVe. In addition, word2vec word groupings can group unique n-gram words than GloVe word groupings. In word2vec word groupings, 90.0% similarity give the best accuracy while the accuracy of 70.0% similarity as high as that and F-measure of 70.0% similarity group achieved the highest result.

The results of classification with SVM for each experiment are shown in Table 6.6. The performance of n-gram-based words without grouping, word2vec word grouping with 90.0%

similarity and GloVe word grouping with 90.0% similarity are similar in both accuracy and F-measure when tf-idf values are applied. However, in the case of using sentiment values, word2vec word groupings with similarity 90.0% achieve the best accuracy and F-measure. Thus, also in this case, word2vec is a better way.

Table 6.5 Classification with Sentiment Values Summation

	Accuracy	**Precision**	**Recall**	**F-measure**
Experiment I (n-gram words)	65.65%	61.15%	85.85%	71.42%
Experiment II (word2vec - 70%)	67.91%	63.51%	84.21%	**72.41%**
Experiment II (word2vec - 80%)	66.50%	62.46%	82.71%	71.17%
Experiment II (word2vec - 90%)	**68.05%**	**64.84%**	78.85%	71.16%
Experiment III (GloVe - 70%)	61.41%	57.19%	**90.74%**	70.16%
Experiment III (GloVe - 80%)	65.24%	60.86%	85.41%	71.07%
Experiment III (GloVe - 90%)	63.74%	59.06%	89.56%	71.18%

6.3.2 Discussion

Since this dataset is used as a bigger dataset similar to Burmese with supervised labels, all the experiments conditions are the same as the Burmese data processing. The total number of extracted n-gram words and unique n-gram words of Amazon reviews data are shown in Table 6.4. For n-gram-based word segmentation (Experiment I), among the n-gram words from the original data, the words that appear at least five times are extracted in each segmentation step. After processing segmentation with all n-gram (from 10-gram to 3-gram), n-gram words that appear more than 500 times are removed (similar to the processing of Burmese news articles). After segmentation and extraction steps, the data still contains a large size of unique n-gram words.

To reduce the number of words in n-gram datasets, I proposed the method to employ distributive word representation models, word2vec and GloVe, to calculate word similarities for grouping words. This is evaluated in Experiment II and III. With the word grouping with word2vec (Experiment II), around 61.0%, 47.0% and 3.0% of unique n-gram words are grouped in cosine similarity of 70.0%, 80.0%, and 90.0% respectively. With GloVe (Experiment III), around 0.1%, 0.04% and 0.003% of unique n-gram words are grouped. This means that word2vec words grouping is more efficient than GloVe words grouping for this dataset.

Table 6.6 Classification with SVM

	SVM	Accuracy	Precision	Recall	F-measure
Experiment I (n-gram words)	Tf-idf values	89.68%	90.31%	88.88%	89.59%
	Sentiment Values	84.83%	85.13%	84.43%	84.77%
Experiment II (word2vec-70%)	Tf-idf values	84.13%	84.14%	84.11%	84.12%
	Sentiment Values	82.45%	82.38%	82.51%	82.43%
Experiment II (word2vec-80%)	Tf-idf values	87.03%	87.30%	86.66%	86.97%
	Sentiment Values	83.95%	84.17%	83.64%	83.89%
Experiment II (word2vec-90%)	Tf-idf values	89.58%	90.09%	**88.96%**	89.51%
	Sentiment Values	**85.46%**	**85.68%**	85.15%	**85.41%**
Experiment III (Glove – 70%)	Tf-idf values	89.11%	89.29%	88.88%	89.08%
	Sentiment Values	83.75%	82.65%	**85.46%**	84.03%
Experiment III (Glove – 80%)	Tf-idf values	89.21%	89.46%	88.90%	89.17%
	Sentiment Values	84.52%	84.15%	85.06%	84.59%
Experiment III (Glove – 90%)	Tf-idf values	**89.78%**	**90.47%**	88.94%	**89.69%**
	Sentiment Values	84.54%	84.89%	84.05%	84.46%

In the first way of classification, word2vec word groupings are better than n-gram-based words without words grouping and GloVe in the case of performance and words grouping. It seems that word2vec is more suitable than GloVe partly because word2vec is based on simple word occurrence within a specific window size while GloVe is based on global word co-occurrence. In this case, the performance of word2vec word grouping with 70.0% similarity can be considered as the best one because it can give the highest F-measure and the accuracy can be compared with the best accuracy. In addition, it can group around 61.0% of the original data which gives the smallest number of unique n-gram words among four types of experiment.

For SVM classifier, training and testing datasets are based on 10-fold cross validation for all experiments. In this dataset, using tf-idf values gives better results than using sentiment values in all experiments. Using tf-idf features is a trustworthy method if the data size is sufficiently large. The performance of word2vec and GloVe word groupings is similar, and a little bit better than n-gram-based words without word grouping.

6.4 Comparison between Two Datasets

To compare a small dataset of Burmese newspaper articles and a large dataset of pseudo-Burmese Amazon product reviews, the performance of using Burmese articles gives better performance in the first way of classification while using Amazon reviews provides a better result in the second way of classification. For calculating words similarity, 61.0%, 47.0% and 3.0% of words grouping with word2vec and 0.1%, 0.04% and 0.003% of words grouping with GloVe in the Amazon reviews data is more than 45.0%, 4.0% and 0.5% of words grouping with word2vec and less than 0.01% of words grouping with GloVe in Burmese news data case. Therefore, both word2vec and GloVe can group n-gram words more effectively in a large dataset of pseudo-Burmese Amazon product reviews.

The number of n-gram word grouping for Amazon data is more than Burmese data. This is because each Burmese news article contains many kinds of words (different features) and unlike English, the Burmese language has many different words even for the same meaning. For example, the word 'I' can be written with variant words in the Burmese lexicon, and for the plural words, English add 's/es' to the original word while Burmese has different words that can be added to the original word to be plural forms. The Burmese language has different forms of words to express the same kind of meaning. Thus, even a small size of data contains many kinds of features.

In both data types, the result of classification with SVM is better than classification with simple sentiment values summation. The former one is a supervised machine learning technique where the classifier is trained with supervised answer data whereas the latter one is performed based on simple calculation, the summation of sentiment values of words.

In all the experiments, the performance of classification also depends on the choice of positive and negative seed words. In this work, I decided to choose the words that appear frequently as a good or bad expression in the text.

6.5 Concluding Remarks

By summarizing the results from all experiments, the proposed approach variable length of n-gram-based words with distributive word representation model can give the similar or higher performance than baseline approach. In addition, the proposed approach does not focus on a specific language property and it performs without using dedicated tools and resources while the baseline approach is specific only to the Burmese language property.

CHAPTER 7

CONCLUSION

7.1 Summary of This Thesis

The goal of this thesis is to propose a method of effective sentiment analysis for low resource languages. Low resource languages often lack tagged datasets, trustworthy word segmentation and other basic natural language processing tools. In particular, the Burmese language as well as Thai, Lao, and Khmer, has no explicit word boundaries in texts, which makes most language processing tasks more difficult than English and other languages, whose writing use spaces to segment words.

My proposal is to employ a variable-length n-gram word model instead of ordinary word segmentation, to use distributive word representation vectors to reduce the number of n-gram words in the data by grouping them according to cosine similarity.

My achievement is summarized as follows. First, for the purpose of preparing a baseline model, a CRF-based word segmentation for the Burmese language is newly proposed, together with Burmese Character Clusters (BCC) based on Thai Character Clusters (TCC). Its accuracy is 98.8%, which is comparable to the state-of-the-art performance of word segmentation for Burmese and Thai, though it is based on a tiny set of texts.

Second, I proposed a method to use a variable-length n-gram word model instead of ordinary word segmentation for sentiment analysis of low resource languages. A variable-length n-gram word model segments text into n-gram words but n is varied. To enable this, the target text is first segmented with the maximum length of n, or N-gram, and frequent N-gram words are selected. Then, the rest of the text is segmented with N-1, and frequent (N-1)-gram words are selected. This procedure is iteratively conducted to the minimum length of n, or M-gram. M and N are manually decided, and this thesis set M and N as 3 and 25, respectively, considering the Burmese writing characteristics. Even when this model is applied without further techniques, sentiment classification by Support Vector Machines achieved similar to or better than when ordinary word segmentation is applied.

Third, as the number of n-gram words in a text becomes much higher than the number of ordinary words in the same text, the employment of distributive word representation model is proposed to reduce the number of n-gram words in a text by grouping n-gram words based

on cosine similarity. As regards distributive word representation models, word2vec and GloVe are evaluated with different rates of cosine similarity. The experiment result tells that word2vec with 80.0% of cosine similarity performed best, which is equal to or better than ordinary word segmentation.

All these evaluations were performed with two datasets. One is a tiny collection of Burmese newspaper articles, and the supervised label was created based on questionnaire surveys to Burmese people. The other is a larger dataset of Amazon Product Reviews, but it is preprocessed to make it a pseudo-Burmese dataset by removing spaces as well as inflections. The latter was done because it is almost impossible to construct a larger dataset of the Burmese language for this purpose, which is another usual issue when coping with low resource languages.

Comparative experiments were conducted with different cosine similarities for two tiny and larger datasets. For the feature set of the input, a usual bag-of-words model is adopted, and each feature was set either tf-idf or sentiment values. Sentiment values were calculated with the target datasets using a method proposed by Turney and Littman. For classification, supervised machine learning technique, SVM, is applied and to train and test the classifier, the original dataset is split into training and test datasets based on 10-fold cross-validation. By splitting this way, all the data is used as a training and test data. For the tiny set, sentiment values performed better than tf-idf while tf-idf performed mostly better for the larger set. In both data types, using tf-idf or sentiment values features gave better results of classification than using the simple addition of sentiment values.

In conclusion, the proposed method of this study shows that a variable-length n-gram word model with similarity-based grouping is a promising method for low resource languages. Since the issue of lack of resources is one of the most serious issues in low resource languages, this study is considered to be a first-step contribution to manage the issue in an efficient way.

7.2 Future Work

The proposed approach can give a good performance of classification for both types of data in this study. Therefore, this work is expected to be employed to many low resource language tasks. For instance, since the structure of Burmese language is similar with Khmer and Lao, the proposed approach can be expected to apply those languages.

For parameter setting in the case of segmentation of variable length of n-gram words, I start to choose from 25 and stop at 3 for the Burmese articles and from 10 to 3 for the Amazon reviews data. This is a kind of heuristic decision. This kind of manually parameter setting

should be able to choose automatically in order to adjust with the use of data type. By automatic adjusting the parameter setting, the performance may be increased.

It would be better if similar experiments are conducted with different languages to evaluate the proposed method. In this study, I used the Amazon reviews data and made this dataset as a pseudo-Burmese dataset. Since it cannot be proved that the data is totally similar with the Burmese language, other languages which have similar characteristics with the Burmese language and have a bigger size of data can also be used in experiments.

REFERENCES

Asghar, M. Z., Khan, A., Ahmad, S., Khan, I. A., and Kundi, F. M. (2015). "A Unified Framework for Creating Domain Dependent Polarity Lexicons from User Generated Reviews." *PLoS ONE*, 10 (10), e0140204.

Aung, K. Z. (2016). "A Lexicon Based Sentiment Analyzer Framework for Student-Teacher Textual Comments." *International Journal of Scientific and Research Publications*, 6(2), 277-280.

Aung, N. T. T., and Thein, N. L. (2011). "Word Sense Disambiguation System for Myanmar Word in Support of Myanmar-English Machine Translation.", *Proc., SICE Annual Conference (SICE)*, Tokyo, Japan, 2835-2840.

Aydogan, E., and Akcayol, M. A. (2016). "A Comprehensive Survey for Sentiment Analysis Tasks Using Machine Learning Techniques.", *Proc., International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, Sinaia, Romania, 1-7.

Aye, Y. M., and Aung, S. S. (2017). "Sentiment Analysis for Reviews of Restaurants in Myanmar Text.", *Proc., International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Kanazawa, Japan, 321-326.

Chea, V., Thu, Y. K., Ding, C., Utiyama, M., Finch, A., and Sumita, E. (2015). "Khmer Word Segmentation Using Conditional Random Fields.", Proc., *In Khmer Natural Language Processing 2015*, Phnom Penh, Cambodia, 1-8.

Chen, Y., and S. Steven. (2014). "Building Sentiment Lexicons for All Major Languages.", *Proc., 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland, USA, 383-389.

Cieri, C., Maxwell, M., Strassel, S., and Tracey, J. (2016). "Selection Criteria for Low Resource Language Programs.", *Proc., Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portoroz, Slovenia, 4543-4549.

Devi, D. V. N., Kumar, C. K., and Prasad, S. (2016). “A Feature Based Approach for Sentiment Analysis by Using Support Vector Machine.”, *Proc., IEEE 6th International Conference on Advanced Computing*, Bhimavaram, India, 3-8.

Ding, C., Thu, Y. K., Utiyama, M., and Sumita, E. (2016). “Word Segmentation for Burmese (Myanmar)”, *ACM Transactions on Asian and Low-resource Language Information Processing*, 15(4), Article No. 22, 1-10.

Du, L., Li, X., Liu, C., Liu, R., Fan, X., and Yang, J. (2016). “Chinese Word Segmentation Based on Conditional Random Fields with Character Clustering.”, *Proc., International Conference on Asian Language Processing (IALP)*, Tainan, Taiwan, 258-261.

Feldman, R. (2007). “Introduction to Sentiment Analysis.” *IJCAI*, <http://ijcai13.org/files/tutorial_slides/tf4.pdf> (April 29, 2018).

Fellbaum, C. (1998). “WordNet: An electronic lexical database.” MIT Press, Cambridge, Massachusetts, United States.

Gamon, M., and Aue, A. (2005). “Automatic Identification of Sentiment Vocabulary: Exploiting Low Association with Known Sentiment Terms.”, *Proc., ACL Workshop on Feature Engineering for Machine Learning in NL*P, Ann Arbor, Michigan, 57-64.

Garcia, E. (2015). “Cosine Similarity Tutorial.” <http://www.minerazzi.com/tutorials/cosine-similarity-tutorial.pdf> (April 18, 2018)

Godsay, M. (2015). “The Process of Sentiment Analysis: A Study.” *International Journal of Computer Applications*, 126(7), 26-30.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). “Deep Learning.”, MIT Press, Cambridge, Massachusetts, United States.

He, R., and McAuley, J. (2016). “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering.”, *Proc., 25th International Conference on World Wide Web*, Montreal, Quebec, Canada, 507-517.

Holzinger, A. (2016). “Introduction to Word Embeddings Word Vectors (Word2Ve/ GloVe) Tutorial.” TU Wien <http://hci-kdd.org/wordpress/wp-content/uploads/2016/06/T2-185A83-WORD-VECTOR-TUTORIAL-VO-2016.pdf> (November 15, 2017)

Htike, K. W. W., Pa, W. P., and Thu, Y. K. (2017). “Myanmar Part-of-Speech Corpus for Myanmar NLP Research and Developments.” <https://github.com/ye-kyaw-thu/myPOS> (May 3, 2018)

Jenny, M., and Tun, S. S. H. (2016). “Burmese A Comprehensive Grammar.” *Routledge*, <http://www.nlb.gov.sg/biblio/202458430> (April 29, 2018)

Katrekar, A. (2014). “An Introduction to Sentiment Analysis.” *GlobalLogic*, < https://www.globallogic.com/il/gl_news/an-introduction-to-sentiment-analysis/> (April 30, 2018)

Kudo, T. (2013). “CRF++: an open source implementation of Conditional Random Fields (CRFs).” <https://taku910.github.io/crfpp/ > (March 10, 2017)

Lafferty, J., Mccallum, A., and Pereira, F. (2001). “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.”, *Proc., 18th International Conference on Machine Learning 2001 (ICML 2001)*, Williams College, Williamstown, MA, USA, 282-289.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). “Introduction to Latent Semantic Analysis.” *Discourse Processes*, 25, 259-284.

Maung, Z. M., and Mikami, Y. (2008). “A Rule-based Syllable segmentation of Myanmar Text”, *Proc., IJCNLP-08 Workshop on NLP for Less Privileged Languages*, Hyderabad, India, 51-58.

McAuley, J., Targett, C., Shi, O., and Hengel, A. (2015). “Image-Based Recommendations on Styles and Substitutes.”, *Proc., 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Santiago, Chile, 1-8.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). “Efficient Estimation of Word Representations in Vector Space.”, Proc., *International Conference on Learning Representations (ICLR 2013)*, Scottsdale, Arizona, USA, 1-12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). “Distributed Representations of Words and Phrases and their Compositionality.”, *Proc., 26th*

International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, 1-9.

Mon, A. M., and Thein, T. (2013). "Myanmar Spell Checker." *International Journal of Science and Research*, 2(1), 333-340.

Mon, P. Y., and Mikami, Y. (2011). "Myanmar Language Search Engine." *International Journal of Computer Science*, 8(2), 118-126.

Myint, C. (2011). "A Hybrid Approach for Part-Of-Speech Tagging of Burmese Texts.", *Proc., 2011 International Conference on Computer and Management (CAMAN)*, Wuhan, China, 214-217.

Myint, P. H., Htwe, T. M., and Thein, N. L. (2011). "Normalization of Myanmar Grammatical Categories for Part-of-Speech Tagging." *International Journal of Computer Applications*, 36(1), 10-17.

Myint, P. H., Htwe, T. M., and Thein, N. L. (2011). "Bigram Part-of-Speech Tagger for Myanmar Language." *Proc., Computer Science and Information Technology*, Singapore, 147-152.

Natural Language Processing Lab, University of Computer Studies, Yangon, Myanmar. (2011). "Myanmar Word Segmentation." <http://www.nlpresearch-ucsy.edu.mm/wordsegmentation.html> (June 13, 2018)

NSS. (2017). "An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec." *Analytics Vidhya*, <https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/> (December 24, 2017).

Okell, J. (2018). "A Corpus of Modern Burmese." *Zenodo*, <https://zenodo.org/record/1202324#.WurCYUxuI2x > (May 3, 2018)

Pa, W. P., Thu, Y. K., Finch, A., and Sumita, E. (2015). "Word Boundary Identification for Myanmar Text Using Conditional Random Fields." *Genetic and Evolutionary Computing*, 388, 447-456.

Pennington, J., Socher, R., and Manning, C. D. (2014). "GloVe: Global Vectors for Word Representations.", *Proc., 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* , Doha, Qatar, 1532-1543.

Phyue, S. L. (2011). “Construction of Myanmar WordNet Lexical Database.”, *Proc., 2011 IEEE Student Conference on Research and Development (SCOReD),* Cyberjaya, Malaysia, 327-332.

Sayad, S. (2010). “An Introduction to Data Science.” <http://www.saedsayad.com/naive_bayesian.htm> (June 20, 2018)

Soe, W., and Theins, Y. (2015). “Syllable-based Myanmar Language Model for Speech Recognition”, *Proc., 14th International Conference on Computer and Information Science,* Las Vegas, NV, USA, 1-13.

Swe, T., and Tin, P. (2005). “Recognition and Translation of the Myanmar Printed Text Based on Hopfield Neural Network”, *Proc., 6th Asia-Pacific Symposium on Information and Telecommunication Technologies,* Yangon, Myanmar, 99-104.

Thant, W. W., Aung, N. T. T., Htay, S. S., Htwe, K., K., and Yar, K. T. (2017). “Assigning Polarity Scores to Facebook Myanmar Movie Comments.” *International Journal of Computer Applications,* 177(6), 24-29.

The Library of Congress. (2011). “Burmese Romanization Table.” <https://www.loc.gov/catdir/cpso/roman.html> (April 29, 2018)

Theeramunkong, T., Sornlertlamvanich, V., Tanhermhong, T., and Chinnan, W. (2000). “Character Cluster Based Thai Information Retrieval.”, *Proc., fifth international workshop on Information retrieval with Asian languages*, Hong Kong, China, 75-80.

Thet, T. T., Na, J. C., and Ko, W. K. (2008) “Word segmentation for the Myanmar language”, *Journal of Information Science*, 34(5), 688-704.

Tongtep, N., and Theeramunkong, T. (2010). “Simultaneous Character-Cluster-Based Word Segmentation and Named Entity Recognition in Thai Language.”, *Proc., 5th International Conference on Knowledge, Information, and Creativity Support Systems*, Chiang Mai, Thailand, 216-225.

Tsvetkov, Y. (2017). “Opportunities and Challenges in Working with Low-Resource Languages.” < http://www.cs.cmu.edu/~ytsvetko/jsalt-part1.pdf > (June 12, 2018)

Turney, P. D. (2002). “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews.”, *Proc., 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, Pennsylvania, 417-424.

Turney, P. D., and Littman, M. L. (2003). “Measuring Praise and Criticism: Inference of Semantic Orientation from Association.”, *ACM Transactions on Information Systems*, 21(4), 315–346.

Vu, X. S., and Park, S. B. (2014). “Construction of Vietnamese SentiWordNet by using Vietnamese Dictionary.”, *Proc., 40th Conference of the Korea Information Processing Society*, Seoul, South Korea, 745-748.

Wai, P. P. (2011). “Myanmar To English Verb Translation Disambiguation Approach Based on Naïve Bayesian Classifier.”, *Proc., 3rd International Conference on Computer Research and Development (ICCRD)*, Shanghai, China, 6-9.

Wai, T. T., and Thein, N. L. (2011). “Markov-based Reordering Model for English-Myanmar Translation.”, *Proc., SICE Annual Conference*, Tokyo, Japan, 2736-2740.

Win, A. T. (2011). “Words to Phrase Reordering Machine Translation System in Myanmar-English Using English Grammar Rules.”, *Proc., 3rd International Conference on Computer Research and Development*, Shanghai, China, 50-53.

Weston, J. (2006). “Support Vector Machine Tutorial.” *Computer Science Columbia University*, <http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf> (April 29, 2018).

Zin, K. K., and Thein, N. L. (2009). “Part of Speech Tagging for Myanmar Using Hidden Markov Model.”, *Proc., International Conference on the Current Trends in Information Technology (CTIT)*, Dubai, United Arab Emirates, 1-6.

Zin, T. T., Soe, K. M., and Thein, N. L. (2011). “Translation Model of Myanmar Phrases for Statistical Machine Translation.” *Proc., 7th International Conference on Advanced Intelligent Computing Theories and Applications: With Aspects of Artificial Intelligence*, Zhengzhou, China, 235-242.

7Day Daily, (Online) Available on http://www.7daydaily.com/ (10 January, 2018)

VITAE

Name Miss Myat Lay Phyu

Student ID 5930223007

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering (Information Science And Technology)	University of Technology (Yatanarpon Cyber City) Myanmar	2016

List of Publications and Proceedings (If Possible)

Phyu, M. L. and Hashimoto, K. (2017). "Burmese Word Segmentation with Character Clustering and CRFs.", *Proc., 14th International Joint Conference on Computer Science and Software Engineering (JCSSE2017),* Nakhon Si Thammarat, Thailand, 1-6.

Phyu, M. L. and Hashimoto, K. (2018). "Sentiment Analysis of the Burmese Language using the Distributive Representation of n-gram-based Word.", *Proc., 4th Joint Symposium on Computational Intelligence (JSCI2018),* King Mongkut's University of Technology Thonburi (KMUTT), Thailand, 1-2.