UNIVERSITY OF TECHNOLOGY (YATANARPON CYBER CITY)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE

STATISTICAL MACHINE TRANSLATION BETWEEN MYANMAR WRITTEN TEXT AND MYANMAR SIGNWRITING

BY

HNIN WAI WAI HLAING

MASTER THESIS

MAY, 2018

PYIN OO LWIN

UNIVERSITY OF TECHNOLOGY (YATANARPON CYBER CITY)
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
DEPARTIMENT OF INFORMATION SCIENCE

STATISTICAL MACHINE TRANSLATION BETWEEN MYANMAR WRITTEN TEXT AND MYANMAR SIGNWRITING

BY
HNIN WAI WAI HLAING

A THESIS
SUBMITTED TO THE FACULTY OF
INFORMATION AND COMMUNICATION TECHNOLOGY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING
(INFORMATION SCIENCE AND TECHNOLOGY)

MAY, 2018
PYIN OO LWIN

ACKNOWLEDGEMENTS

First of all, I would like to express my special thanks to Dr. Aung Win, Rector, University of Technology (Yatanarpon Cyber City), for initiating the Master Degree Programme at the University of Technology (Yatanarpon Cyber City).

I would like to express my respectful gratitude to Dr. Soe Soe Khaing, Pro-rector, University of Technology (Yatanarpon Cyber City), for her kind permission to complete this thesis. I am thankful to her for giving enough consideration to her ideas and views.

I would like to express my grateful thanks to Dr. Hnin Aye Thant, Professor and Head, Department of Information Science, Faculty of Information Science and Communication Technology, University of Technology (Yatanarpon Cyber City), for her kind guidance, encouragement in making my thesis to complete successfully.

I am very grateful to Dr. Phyu Phyu Tar, Professor and Course Coordinator of Information Science and Technology, Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), for her arrangement, valuable suggestion, comments and advice in completion of thesis.

I am very grateful to my supervisor, Dr. Nandar Win Min, Assistant Lecturer, Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), for her kind guidance and encouragement. She supports and guides a lot in this thesis, particularly, at the level of quality of presentation.

I would like to express my special thanks to my co-supervisor, Dr. Ye Kyaw Thu, Researcher, Artificial Intelligence Lab., Department of Information and Communication Engineering, Faculty of Computer Science and Systems Engineering, Okayama Prefectural University, Japan, for his patient guidance, suggestion, comments and advice. He has been very supportive in this thesis, and also guides a lot in completion of this thesis.

I am also thankful to my co-supervisor, Daw Swe Zin Moe, Assistant Lecturer, Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), for her kind guidance, helps and encouragement.

Very Special thanks to Daw Khaing Khaing Oo, Lecturer and Head of English Department, University of Technology (Yatanarpon Cyber City) for her valuable supports from the language point of view in my thesis work.

I would like to thank principals, teachers, SL trainers and students of School for the Deaf (Mandalay), Mary Chapman School for the Deaf Children (Yangon) and School for the Deaf, Tarmwe (Yangon), Myanmar Deaf Society and Literacy and Language Development for the Deaf for their kind contribution to our research. This research is partially supported by Ministry of Education, Department of Higher Education (Myanmar).

I would like to thank my friend, Hlaing Myat Nwe, Master of Engineering (Information Science and Technology), University of Technology (Yatanarpon Cyber City), for her helpful works in data collection and encouragement.

I would like to thank a lot to all my teachers for their mentoring, encouragement, and recommending this dissertation.

Finally, I am grateful to my parents and friends who specially offered strong moral and physical support, care and kindness, during the year of my thesis study.

ABSTRACT

This research introduces the first evaluation of automatic machine translation between Myanmar Written Text (MWT) and Myanmar SignWriting (MSW). The main motivation is to introduce SignWriting to the Myanmar Deaf society with the help of statistical machine translation because they face various difficulties in communicating with hearing people and there are limited resources of information written in their languages. In this thesis, the current developing parallel corpus is only focused on the emergency domain (e.g., fires, earthquake, floods, storms, accidents, police, health, number, date and time). This corpus contains Myanmar Written Text (MWT), the transcribed Myanmar Sign Language (MSL) and its equivalent Myanmar SignWriting (MSW). The statistical approaches were applied in this thesis: phrase-based, hierarchical phrase-based and the operation sequence model. Three different segmentation schemes were studied: syllable segmentation, word segmentation for MWT and sign unit based segmentation for MSL. In addition, three translation experiments were done between MWT-MSW, MWT-MSL and MSL-MSW and in both directions. The performance of the machine translation systems was automatically measured in terms of BLEU and RIBES for all experiments. According to the overall experimental results, MWT-MSW translation achieved the lower score than the MWT-MSL and MSL- MSW translations because of its grammar structure. 10-fold cross validation results produced promising results even with the limited training data and this can be developed into a useful machine translation system as more data becomes available in the future.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

BLEU	BiLingual Evaluation Understudy
HPBSMT	Hierarchical Phrase-Based Statistical Machine Translation
ISWA	International SignWriting Alphabet
MF	Manual Feature
MSL	Myanmar Sign Language
MSW	Myanmar SignWriting
MT	Machine Translation
MWT	Myanmar Written Text
NMF	Non-Manual Feature
NMT	Neural Machine Translation
OSM	Operation Sequence Model
PBSMT	Phrase-Based Statistical Machine Translation
RIBES	Rank-Based Intuitive Bilingual Evaluation Score
SMT	Statistical Machine Translation
SL	Sign Language
SWML	SignWriting Markup Language
WER	Word Error Rate

CHAPTER 1
INTRODUCTION

This thesis contributes the machine translation between Myanmar Written Text (MWT) and Myanmar SignWriting (MSW) using the statistical approaches. This chapter introduces about the overall explanation of sign language, history of SignWriting, machine translation strategies, aim and objectives, contributions and the scope of this thesis.

1.1. Introduction to Sign Language

Sign Languages are natural language developed by Deaf and used in everyday life. Wherever vocal communication is impossible, sign language can be used as a bridge between Deaf and hearing people. However, sign languages are not at all uniform according to the culture and environments. Different countries use different sign languages. Signers may have many difficulties to understand other signers from different countries because of the sign language grammar structure and usage. Nowadays, there perhaps three hundred sign languages are in use around the world: American Sign Language, British Sign Language, Swedish Sign Language and South African Sign Language, etc.

According to the 2014 Myanmar national census, there are about 4.6% of the populations are disabling and 1.3% of the population are deaf and hearing impairment in Myanmar [14Cen]. Myanmar Sign Language (MSL) is a natural language, a primary communication for Myanmar Deaf community. Myanmar sign language has also its own grammar structure, which is very different with Myanmar written text. A number of written systems for representing sign languages have been developed, and SignWriting symbols are defined for each sign in every country. In the next section, the brief history of SignWriting will be introduced.

1.2. History of SignWriting (SW)

Every deaf cannot read and write the native writing system grammatically. SignWriting is a system of writing sign languages. It is usually expressed in the shapes of the characters, which are abstract pictures of the hands, face, body and movements of signers. Valerie Sutton [74Val] proposed SignWriting in 1974. At first, she is trying to note all dances, all mime and gesture. It is also called movement-writing system for writing all dances, all sports, and all movements. There are many types of writing system for sign language, such as Hamburg Notation System (HamNoSys), SignWriting, Stoke notation, and so on [00Mar]. Today, it is becoming SignWriting for Deaf communities. In chapter three, the detail explanation of sign language and SignWriting will be described.

1.3. Machine Translation (MT)

Machine translation is automated translation or translation carried out by computer. This process is referred to as natural language processing which uses a bilingual data set and other language assets to build language and phrase models used to translate text. There are many different types of machine translation approaches, and the most widely used are Statistical Machine Translation (SMT), Rule-Based Machine Translation (RBMT), hybrid machine translation and Neural Machine Translation (NMT).

Myanmar sign language machine translation approach needs to prepare bilingual data. There are facing more challenges in building parallel corpus between MWT, MSL and MSW due to the following reasons:

- Myanmar language is under-resourced language, and therefore it does not contain tokenized characters (space, full stop, comma, etc.). Tokenization is used as manual word segmentation.
- There is lack of Myanmar sign languages data collected.
- There are no previously defined SignWriting symbols for Myanmar sign language.

Because of these challenges, there is no machine translation between Myanmar Written Text (MWT) and Myanmar SignWriting (MSW). Considering all the prospects, this research mainly used the three Statistical Machine Translation (SMT) approaches. They are as follows:

- Phrase-Based Statistical Machine Translation (PBSMT)
- Hierarchical Phrase-Based Statistical Machine Translation (HPBSMT)
- Operation Sequence Model (OSM).

1.4. Aim and Objectives

The aim and objectives of my thesis are as follows:

- To learn machine translation between MWT-MSW, MWT-MSL and MSL-MSW
- To develop MWT, MSL and MSW parallel corpus
- To measure machine translation performance using Statistical Machine Translation approaches
- To introduce SignWriting to the Myanmar Deaf society
- To fulfill the communication requirements between Deaf and hearing people

1.5. Contribution

Machine translation is a very wide research area in Natural Language Processing (NLP). However, there is lack of research in machine translation for Myanmar sign language. There are many challenges and difficulties as previously described in section 1.3. This research tends to develop parallel text corpus Myanmar Written Text the transcribed Myanmar Sign Language and Myanmar SignWriting. This will investigate machine translation performance using statistical machine translation approaches. Then, it can analyze detail errors (word error and confusion pairs in this thesis) from the translation results. It may be the first study of statistical machine translation between Myanmar Written Text (MWT) and Myanmar SignWriting (MSW). Then, there are a few number of sign trainers in Myanmar Deaf community, who can understand both Myanmar text and sign language. This can be a big difficulty in communication among Deaf and hearing people. For this reasons, this research may lead to be easy communication among Deaf and hearing people, and can acquire good knowledge from reading SignWriting text symbols.

1.6. Organization of the Thesis

This thesis consists of seven chapters as follows:

Chapter one includes the brief introduction of sign language, History of SignWriting, machine translation, aim and objectives, contribution and the scope of the thesis.

Chapter two describes the literature reviews on the previous research paper.

Chapter three presents the detail explanation of Sign Language and SignWriting.

Chapter four highlights three statistical approaches of machine translation theoretical background.

Chapter five describes the building the parallel corpus, the design and implementation of the machine translation.

Chapter six expresses the experimental results of the comparative study of SMT approaches.

Finally, chapter seven includes the conclusion and further study of this research.

1.7. Summary

This thesis tends to reduce the gap between the Deaf and hearing people of the country because there are limited resources of information written in their language. For these reasons, it contributes the first evaluation of machine translation performance using SMT approaches. This chapter described the brief history of sign language and SignWriting, objectives of this thesis. The detail explanation will be described in the next chapters.

CHAPTER 2
LITERATURE REVIEW

Machine translation (MT) is a very wide research area in natural language processing and computational linguistic. The translation from one language to another had been described and published in most research papers. However, machine translation from text to SignWriting is not well-investigated in the literature. Furthermore, previous works in the field of text to Myanmar SignWriting translation is not yet described. This section highlights some of the previous researches relating with sign language, machine translation and segmentation schemes.

2.1. Related Works with Machine Translation

Win Pa Pa et al. [16Win] mentioned an empirical study of five state-of-the-art machine translation to the translation of the low resource languages. The methods applied were phrase-based, hierarchical phrase-based, operation sequence model, string-to-tree and tree-to-string statistical machine translation methods between English (en) and the under-resourced languages Laos (la), Myanmar (mm) and Thai (th) in both directions. They used four languages from the ASEAN-MT parallel corpus in travel domain. It contains six categories: people (greeting, introduction and communication), survival (transportation, accommodation and finance), food (food, beverage and restaurant), fun (recreation, travelling, shopping and nightlife), resources (number, time and accuracy) and special need (emergency and health). They used 20,000 sentences for training, 500 sentences for development and 300 sentences for evaluation. The segmented source and target language were aligned with GIZA++. The alignment was symmetrized by grow-diag-final-and heuristic. The lexicalized reordering model was trained with the msd-bidirectional-fe option. SRILM is used for training the five-gram language model and the minimum error rate training (MERT) was used to tune the decoder parameters. The decoding was done using the Moses decoder.

Ye Kyaw Thu et al. [16Yek] contributed the first large scale evaluation of the quality of automatic translation between Myanmar and twenty other languages, in both directions. They used three different statistical machine translation approaches: phrase-based, hierarchical phrase-based, and the operation sequence model (OSM). In addition, three different segmentation schemes for Myanmar were syllable segmentation, maximum matching word segmentation with a dictionary and supervised word segmentation. They used twenty languages from the multilingual Basic Travel Expressions Corpus (BTEC). In the experiments, 457,249 sentences were used for training, 5,000 sentences for development and 3,000 sentences for evaluation. The results show that the highest quality machine translation was attained with supervised word segmentation in all of the experiments. For almost all language pairs the HPBSMT approach gave the highest translation quality when measured in terms of both the BLEU and RIBES scores.

Jan Bungeroth and Hermann Ney [04Jan] proposed a statistical machine translation system for Sign Language and written language, especially for the language pair German Sign Language (DGS) and German. They also introduced the statistical machine translation, notation systems for Sign Language and the corpus preparation. They used statistical models like IBM Model 1-4 and others like Hidden Markov Models (HMM). And then, they compared word error rate (WER) and position-independent word error rate (PER).

Chenchen Ding et al. [14Che] conducted dependency-based head finalization for statistical machine translation (SMT) for Myanmar (Burmese). Their approach is a combination of two approaches. The first is a head-driven phrase structure grammar (HPSG) based head finalization for English-to-Japanese translation, the second is dependency-base pre-ordering originally designed for English-to-Korean translation. They experimented on Chinese-, English-, and French-to-Myanmar translation, using a statistical preordering approach as a comparison method.

2.2. Related Works with Segmentation

Ye Kyaw Thu et al. [13Yek] examined various word segmentation schemes and their effect on the translation from Myanmar to seven other languages. They performed experiments based on character segmentation, syllable segmentation, human lexical/phrasal segmentation, and unsupervised/supervised word segmentation. The segmentation methods used in the experiments are syllable breaking, maximum matching, unsupervised and supervised approaches. The results show that the highest quality machine translation was attained with syllable segmentation. The Myanmar source segmented by each of the segmentation methods is aligned to the word segmented target languages (Japanese, Hindi, English, Thai, Chinese and Arabic) using GIZA++. Language modeling is done using the IRSTLM and the minimum error rate training (MERT) was used to tune the decoder’s parameters and the decoding is done using the phrase-based SMT system MOSES.

Khin War War Htike [17Khin] proposed the comparison of Six POS Tagging Methods on 10K Sentences Myanmar Language (Burmese) POS Tagged Corpus. A robust Myanmar Part-of-Speech (POS) tagger is necessary for Myanmar natural language processing research. For this reason, an annotated ten thousand sentence POS tagged corpus were manually developed for the general domain. Six POS tagging approaches were also evaluated: Conditional Random Fields (CRFs), Hidden Markov Model (HMM), Maximum Entropy (MaxEnt), Support Vector Machine (SVM), Ripple Down Rules-based (RDR) and Two Hours of Annotation Approach (i.e. combination of HMM and Maximum Entropy Markov Model) on our developed POS tagged corpus. The POS tagging experimental results were measured with accuracy and also manual checking in terms of confusion pairs. The result shows that RDR approach give the best performance for open test-sets. The HMM and MaxEnt approaches also give strong results.

2.3. Related Works with Sign Language

Antonio and Gracliz [02Ant] proposed the SignWriting system and its importance as a deaf sign language writing system. They discussed on the state of the work being done on the development of SignWriting Markup Language (SWML) and SignWriting-based web applications using SWML to render sign languages.

Achraf Othman and Mohamed Jemni [11Ach] designed a statistical machine translation from English text to American Sign Language (ASL). There are two steps in this paper. First, they translate the input text to gloss, a written form of ASL. Second, the output is passed to the WebSign Plug-in to play the sign. The system is mainly based on Moses tool with some modifications and the results are synthesized through a 3D avatar for interpretation. At first, they prepared parallel data. The parallel corpus is composed by a pair of sentences English and American Sign Language (ASL) in a text file. ASL is annotated by gloss. Glosses are written words, where one gloss represents one sign. For word alignment, they used the GIZA++ statistical word alignment toolkit. This tool extracted a set of high-quality word alignment from the original unidirectional alignments sets. This step includes a string matching algorithm. For statistical machine translation (SMT) decoder, MOSES is used in this paper.

Daniel Stein et al. [06Dan] used a phrase-based SMT system for the language pair German and German SL (DGS). The SL corpus is annotated with glosses, including all important grammar features. Their research is focused on morpho-syntactic pre and post-processing enhancement. In the pre-processing step, German is analyzed by a parser, and part-of-speech (POS) information is used to transform nouns into stem forms, split compound words and delete German POS not used in DGS. In the post-processing step, marked positions of discourse entities are added from a database. Some deleted information about emphasis and comparative degree is added as well. Therefore, morpho-syntactic information is not used during the translation process.

Guylhem Aznar and Patrice Dalle [05Guy] presented a solution using an entry process to resolve these variabilities. The proposed solution takes advantage of user interaction to replace the user-entered sign by a standard sign, coming from a dictionary. Through this process, the user choice of symbols is driven by a menu putting forward the most frequent symbols. Their positioning is eased by an automatic positioning of the chosen symbol. The advantage is thus expectable time savings for users. Statistical studies and their necessary algorithms, upon which probability trees are based, will be studied separately.

Sara Morrissey and Andy Way [07Sar] discussed the development of an automatic machine translation (MT) system for translating spoken language text into signed languages (SLs). The applied research area is the airport information announcements for Deaf and hard of hearing people. During the translation process, the decoder takes in an input sentence, searches its three alignment databanks for candidate matches on a sentential, sub sentential chunk and word level. MOSES is used to deduce the phrase for translation, which is then produced as output. The system used two EBMT techniques to improve on the baseline SMT system. The first, ‘chunking method one’, uses the Marker Hypothesis to segment both the source and target data and the resulting chunks and alignments were added to the system. The second, ‘chunking method two’, takes into account the natural lack of closed class lexical items in SLs and segments the ISL data so that each ISL word forms its own chunk which is then aligned with English chunks that have been derived using the Marker Hypothesis. The system is performing for the test data of 118 sentences. They used word error rate (WER) and position-independent word error rate (PER) to calculate distance. A lower percentage score indicates better translations.

Alex Becker et al. [16Ale] described the building an online tool for manually annotating texts in any spoken language with SignWriting in any sign language. They built sign corpus annotation tool, which was implemented in the Java Web platform using the JSF framework (Java Server Faces) and MVC architecture (Model-View-Controller). This system requires two main packages: dictionary and corpora. The first is responsible for controlling the dictionaries and their entry pairs, and a frequency attribute need to maintain for improving candidate sign suggestion. The second is responsible for keeping the corpora and the annotation statuses of each document, where the tokens of each sentence of a document are linked to dictionary entries. After the user creates or selects a raw document, the system runs the document preparation process and segments the document into sentences and then into tokens. Raw documents may be either manually uploaded as text or automatically fetched from Wikipedia given their URL. SignMaker is used as an intermediate tool to produce formal SignWriting.

Antônio and Graçaliz [02Ant] introduced a SignWriting-based approach to Sign Language processing. They mentioned SignWriting Markup Language as a writing system. It is an XML-based language for encoding sign language texts, written in SignWriting, in an application and computer platform independent way. Thus, sign language texts, written in SignWriting and encoded in SWML, can be entered as input to – and also got as output from - any kind of computer program performing any kind of language and document processing (storage and retrieval, analysis and generation, translation, spell-checking, search, animation and dictionary automation, etc.).

2.4. Summary

This chapter has described the previous works related with machine translation, sign language and segmentation schemes. Based on the finding in the previous researches, this thesis evaluates the machine translation performance between Myanmar written text and Myanmar SignWriting. The details of sign language and SignWriting will be explained in the next chapter.

CHAPTER 3
SIGN LANGUAGE AND SIGNWRITING

This chapter describes the explanation of sign language and SignWriting. This is important to understand in detail because sign language and SignWriting are taken part in the preparation of parallel corpus for this thesis.

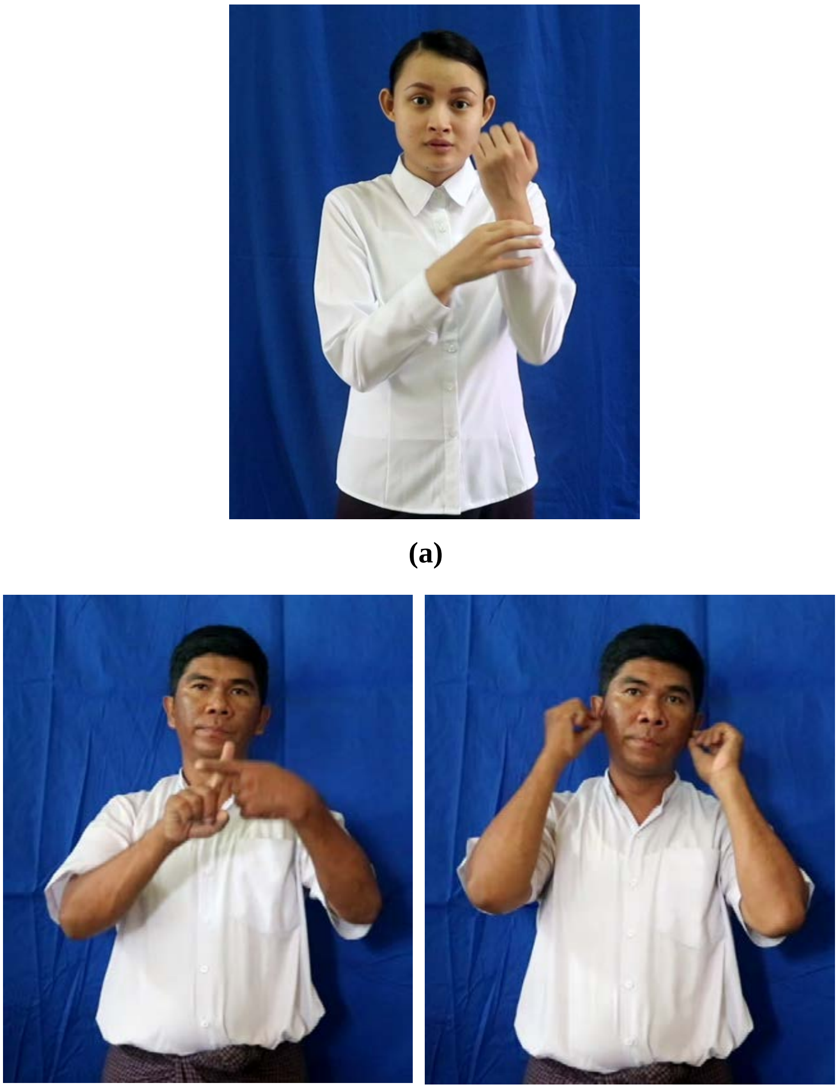
3.1. Sign Language (SL)

SL is the native language of the Deaf community and they can express their needs and the formation of concepts by combining hand shapes, orientation and movement of the hands, arms or body and facial expressions. Deaf uses sign language as primary language to communicate with each other and to get good knowledge. Sign languages are quite difference from one country to country or region to region. Sign languages are at the core of deaf culture. Each may have a syntax and grammar that differs dramatically from the language spoken locally. As an example, the grammatical structure of American Sign Language and Arabic Sign Language are not the same.

It is not clear how many sign languages there are. Each country has its own, native sign language, and some have more than one. The 2013 edition of Ethnologue lists 137 sign languages in world [13Lew]. Although sign language differs in different regions, it mainly depends on the basic parts of sign. It consists of manual features (MFs) and non-manual features (NMFs). Manual features (MFs) are signs performed by one or both hands in different shapes, locations, movements and orientations to express meanings. Non-manual features (NMFs) contain various facial expressions, head tilting, and shoulder raising, mouthing and similar signals that add to hand sign to describe meanings. And then, it grammatically includes questions, negation, relative clauses, and boundaries between sentences [07Man].

3.2. Myanmar Sign Language (MSL)

Almost 1.3 percentage of population in Myanmar are Deaf or hard of hearing. They use its own sign for Deaf community [14Cen]. In this case, the expression and usage of sign language is also slightly different in Myanmar. There are four schools for the deaf in Myanmar, the Mary Chapman School for the Deaf children in Yangon (est. 1904), the School for the Deaf, Mandalay (est. 1964), the Immanuel School for the Deaf in Kalay (est. 2005), School for the Deaf, Yangon, Tarmwe (est. 2014). The sign languages used in Yangon and Mandalay is different. The different signs used in Yangon and Mandalay are shown in Fig 3.1.



(a)

(b)

Figure 3.1. Example of Different Sign used in Yangon and Mandalay (a) "Doctor" Sign of Yangon Sign Language (b) "Doctor" Sign of Mandalay Sign Language

3.3. SignWriting (SW)

SignWriting is one of the notation systems for sign language. It is also the visual representation of sign language. In SignWriting, a combination of iconic symbols for hand shapes, orientation, body locations, facial expressions, contacts and movement are used to represent words in sign language. SignWriting is an alphabet – a list of symbols used to write any sign language in the world. The SignWriting alphabet can be compared to the alphabet used to write English. The same basic alphabets are used to write Danish, German, French and Spanish. In the same way, the symbols in the SignWriting alphabet are international and can be used to write American Sign Language and British Sign Language.

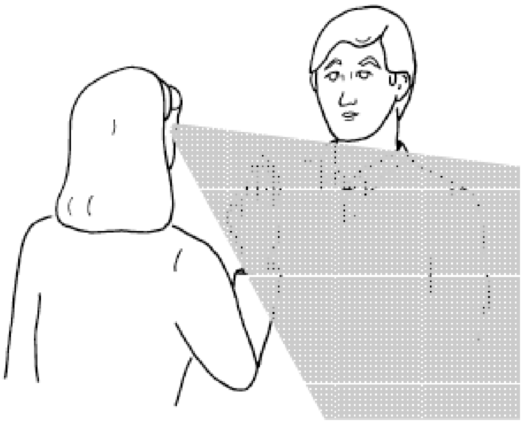


Figure 3.2. Signer's Point of View for Sign Language [10Val]

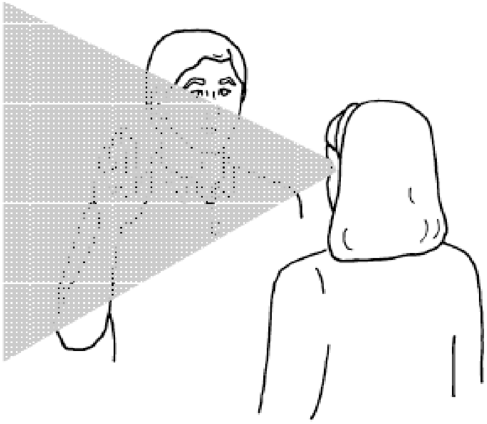


Figure 3.3. Observer's Point of View for Sign Language [10Val]

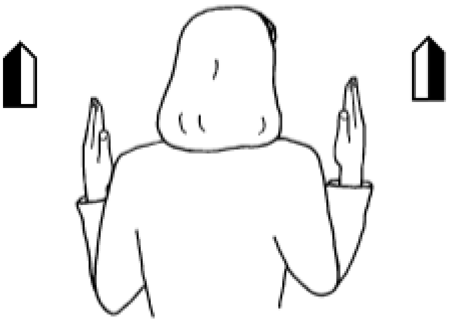


Figure 3.4. Representation of the Hand with SignWriting Symbols [10Val]

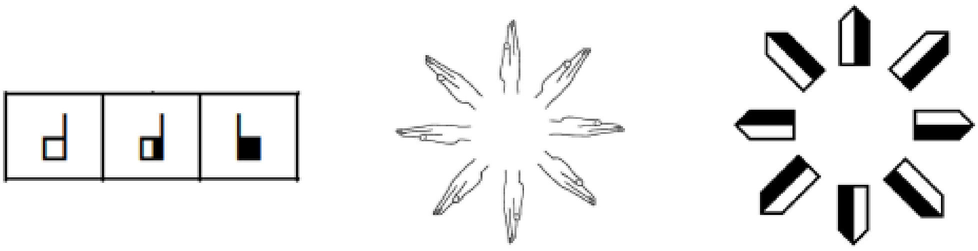
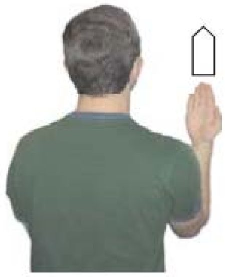
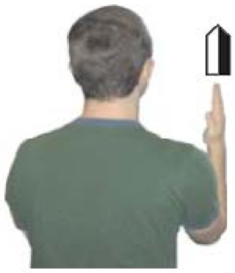
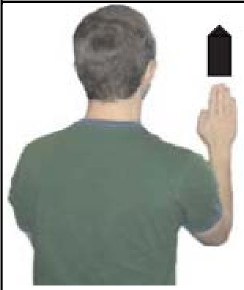


Figure 3.5. (a) Three Filling Symbols (b) Eight Different Rotation of Hand (left) and SignWriting (right) [10Val]

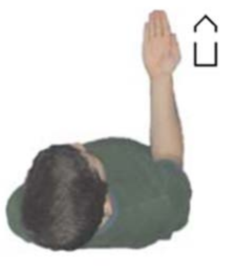
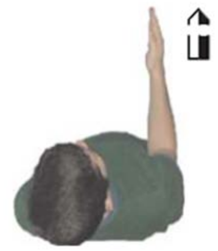
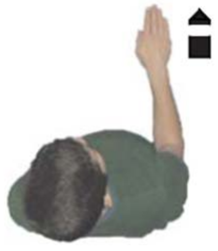
SignWriting represents two points of view: signer's point of view and observer's point of view, as shown in Fig 3.2 and Fig 3.3. However, almost all publications use the signer's point of view, and assume the right hand is dominant. SignWriting symbols are used in a pictograph called a sign box, and written horizontally (left-to-right) or vertically (top-to-bottom).

Table 3.1. Example of SignWriting HAND-FLAT Hand Shapes with Parallel Wall Plane (FRONT view) [10Ada]

Front View		• Palm of the hand • White or hollow symbol • the palm faces towards your body
		• The side of the hand with half-white and half-dark symbol • The half-white section shows the direction of the palm and the half-dark section represents the back of the hand
		• Back of the hand • Black or filled-in symbol • The black symbol shows that the back of the hand faces towards your body

These symbols can be rotated in eight directions and place anywhere in the writing area, as shown in Fig 3.4 and Fig 3.5 [10Val]. Hand orientation is also important for SignWriting and there are three different filling symbols and eight different spatial rotation symbols for each symbol.

Table 3.2. Example of SignWriting HAND-FLAT Hand Shapes with Parallel Floor Plane (TOP view) [10Ada]

Top View		• The hand is parallel to the floor. • looking down at the palm from overhead • The white symbol with space is at the knuckle joint
		• Side of the hands seeing from overhead • Half-white and half-black symbols with a space at the knuckle joint
		• Back of the hand • Black or filled-in symbols with a space at the knuckle joint

The orientation of the palm is indicated by filling in the glyph for the hand shape. A hollow outline (white) glyph indicates that one is facing the palm of the hand; a filled (black) glyph indicates that one is facing the back of the hand, and split shading indicates that one is seeing the hand from the side. If an unbroken glyph is used, then the hand is placed in the vertical (wall or face) plane in front of the signer. A band erased across the glyph through the knuckles shows that the hand lies in the horizontal plane, parallel to the floor [10Sut]. The brief description of SignWriting Hand-flat hand shape seeing from front and top views are shown in Table 3.1 and Table 3.2. International SignWriting Alphabet (ISWA) 2010 defines seven categories, 30 groups of symbols to form 652 base symbols and 35,023 final symbols.

Each SignWriting symbol is represented as Unicode. The category one, “Hand”, contains 261 base symbols categorized for 10 groups. The category two, “Movements”, includes 242 base symbols for contact symbols, finger movements, straight movement, curved movement, and circle. The third category, “Dynamic and Timing”, is composed of eight base symbols that are mostly used with movement symbols and punctuation symbols. Dynamic symbols are used to give the feeling or tempo to movement. It combines with punctuation symbols to become the equivalent exclamation points and with contact symbols to add feeling to the facial expression. Timing symbols are used to show alternating or simultaneous movement. The category four, “Head and Faces”, includes 110 base symbols to describe the head movement and head position.

The fifth category, “Body”, contains 18 base symbols, in which torso movement, shoulder, hips and limbs are used in sign language as part of grammar, especially when describing conversations between people, call role shifting, or making spatial comparison between items on the left and items on the right. The symbols of body are important when writing sign language storytelling and poetry. Category six, “Detailed Location”, contains eight base symbols that are used when writing signs on a basis. The symbols in this category are only used in computer software to assist in giving further details for sorting large sign language dictionaries that are sorted by SignWriting symbols. These symbols can help decide which sign should come first and which should come second, in the dictionary.

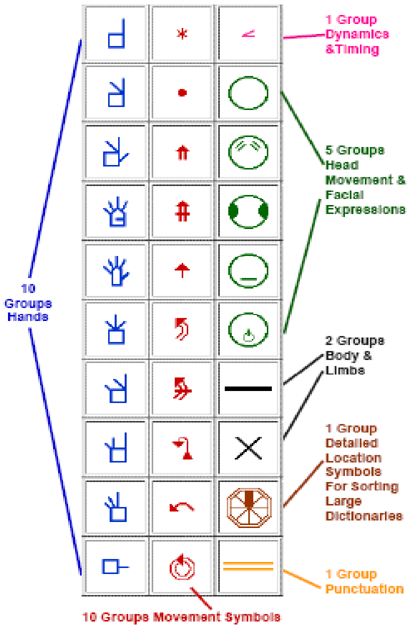


Figure 3.6. Seven Categories of International SignWriting Alphabet 2010 [10Sut]

Finally, category seven, “Punctuation”, contains five base symbols that are used when writing complete sentences or documents in SignWriting. The Punctuation Symbols do not look like the symbols for punctuation in English, but they do have similar meanings [10Ada]. The seven categories of International Signwriting Alphabet (ISWA) 2010 are shown in Fig 3.6.

3.4. Summary

This chapter introduced the nature of sign language, SignWriting and how to use SignWriting symbols in data preparation. In the next chapter, parallel data preparation and its implementation will be explained.

CHAPTER 4

THEORETICAL BACKGROUND

This thesis mainly bases on the statistical approaches in MWT-MSW, MWT-MSL and MSL-MSW translation. In this chapter, the theory applied in this thesis will mainly focus in later sections.

4.1. Machine Translation

Machine translation is linguistically motivated because it aims at achieving the most appropriate translation from one language to other, preserving the meaning of the input text, and producing fluent text in the output language. This means that a MT system will attain success only after it attains natural language understanding. While machine translation is one of the oldest subfields of artificial intelligence research, the recent shift towards large-scale empirical techniques has led to very significant improvements in translation quality, as shown in Fig 4.1. There are many different types of MT approaches, the most widely used in machine translation approaches are Rule-Based Machine Translation (RBMT) and Statistical Machine Translation (SMT). This thesis will focus only on Statistical Machine Translation (SMT) approaches.

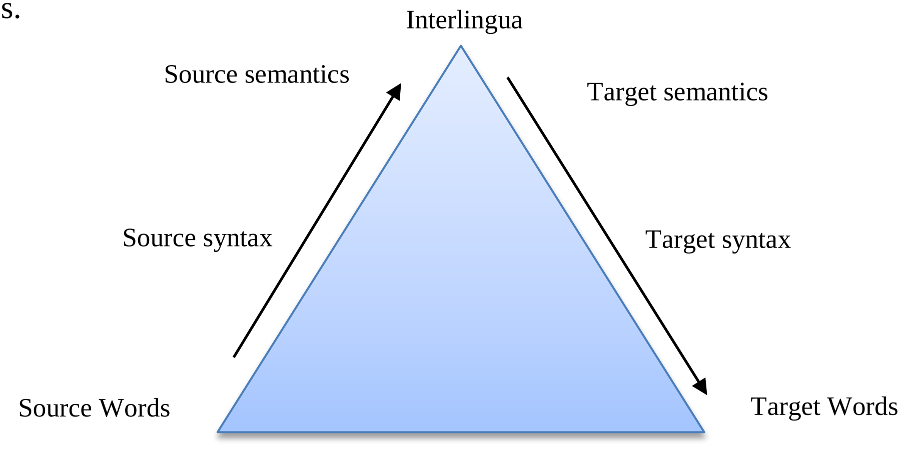


Figure 4.1. Machine Translation Pyramid (Statistical Machine Translation, Koehn)

4.2. Statistical Machine Translation (SMT)

Statistical machine translation utilizes statistical translation models generated from the analysis of monolingual and bilingual training data. Essentially, this approach uses computing power to build sophisticated data models to translate one source language into another. The translation is selected from the training data using algorithms to select the most frequently occurring words or phrases.

Building SMT models is a relatively quick and simple process which involves uploading files to train the engine for a specific language pair and domain. A minimum of two million words is required to train an engine for a specific domain; however, it is possible to reach an acceptable quality threshold with much less. SMT technology relies on bilingual corpora such as translation memories and glossaries to train it to learn language pattern and are uses monolingual data to improve its fluency. SMT engines will prove to have a higher output quality if trained using domain specific training data such as; medical, financial or technical domains.

The statistical methods are most suitable for languages with no rules. The idea behind statistical machine translation comes from information theory. A document is translated according to the probability distribution P(e|f) that a string e in target language (for example, MWT) is the translation of a string in the source language (for example, MSW) in equation 4.1. This is done by building statistical models to represents translation process and find,

$$\hat{e} = \text{argmax}\frac{P(f|e)P(e)}{P(f)} = \text{argmax}P(f|e)P(e) \tag{4.1}$$

where P(f | e) is a translation model that estimate the translation probability and requires training on bilingual parallel text, P(e) is a language model that is trained over large monolingual corpora in the required language, since this allows us to represent the language as accurately as possible as shown in Fig 4.2.

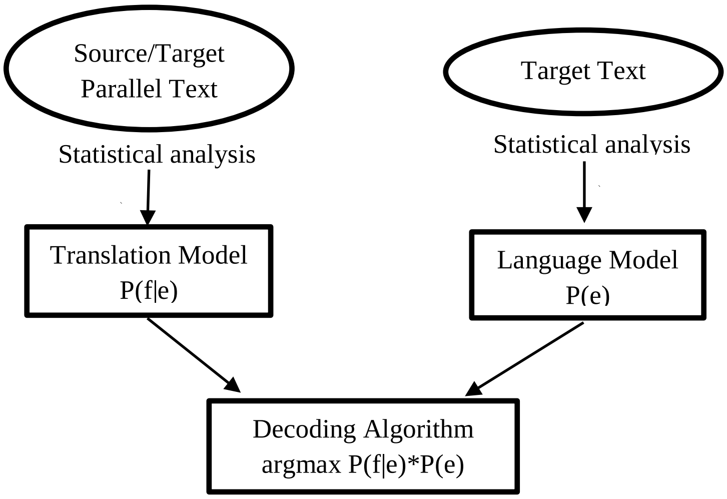


Figure 4.2. Statistical Machine Translation Architecture

A SMT system is characterized by its use of statistical decision theory and Bayes decision rule to minimize the number of decision errors. The statistical approach has several advantages. Statistical decision theory is a well-understood area which provides a sound way to combine several knowledge sources into a global decision criterion with the goal of minimizing the number of errors. It provides a sound framework to resolve the ambiguities of translating the source language text into the target language. The model parameters are estimated from training data. As more and more training data becomes available, SMT systems get better and better. There exists freely available software to build SMT systems. Also, in recent public evaluations, SMT systems performed very well.

Once the Bayes decision rule was specified for statistical machine translation, it has to address three problems [02Och]:

- the **modeling problem**: how to structure the dependencies of source and target language sentences;
- the **search problem**: how to find the best translation candidate among all possible target language sentences;
- the **training problem**: how to estimate the free parameters of the models from the training data.

Several SMT approaches are applied in the natural language processing research areas. In this work, the three SMT approaches are mainly focused: Phrase-Based Statistical Machine Translation (PBSMT), Hierarchical Phrase-Based Statistical Machine Translation (HPBSMT) and Operation Sequence Model (OSM).

4.3. Log-linear Model

The current dominant approach is to use a log-linear combination of feature functions to model directly the posterior probability P(e|f). M feature functions are defined h_m (e, f),m =1, . . . ,M, and the probability of generating e from f is given by Λ in the following equation 4.2.

$$
\begin{aligned}
P_{\Lambda}(e\,|f) &= \frac{\exp(\sum_{m=1}^{M} \lambda_m h_m(e,f)}{\sum_{e'} \exp(\sum_{m=1}^{M} \lambda_m h_m(e',f)} \\
&= \frac{\exp(\Lambda^T h(e,f))}{\sum_{e'} \exp(\Lambda^T h(e',f))} \qquad (4.2)
\end{aligned}
$$

where each feature has a corresponding feature weight λ_m. The equation $\Lambda = (\lambda 1 \ldots \lambda_M)^T$ is for the vector of feature weights and h is for the vector of features. The process of finding the best translation (i.e. the one with the highest probability) is known as decoding. The translation ê maximizes P_Λ (e | f) in the equation 4.3.

$$\hat{e} = \operatorname{argmax}_e \Lambda^T h\,(e, f) \quad (4.3)$$

The decoding process is shown graphically in Fig 4.2. Note that this is equivalent to the source-channel model if we set M = 2, $\lambda_1 = \lambda_2 = 1$ and use feature functions

$$h_1(e, f\,) = \log P(e)$$
$$h_2(e, f\,) = \log P(f\,|e) \quad (4.4)$$

Log-linear models were first used for machine translation [02Och].

4.3.1. Generation of Hypotheses and Decoding

Hypothesis sentences are generated from the input sentence by a generative translation model that depends on the particular translation system being used. Any sequence of words in the output language could be hypothesize and allowed the model to decide on the best translation, the sequences could be reasonably expected to be translations based on what has been seen in the training data. Therefore, rules or phrase tables can be learnt from the parallel text. For a given input sentence, output words have been seen in the parallel text as translations of the input words.

A large problem in the decoding process is the computational complexity. The search space is the set of possible hypotheses during translation and can become very large for a number of reasons. The words in the output sentence are not a direct translation of any words in the input sentence; therefore, the models must allow the insertion and the deletion of input words with no direct translation. Words and phrases in the input sentence may translate in many different ways; algorithms must be devised to find the best hypotheses in the search space efficiently. Decoding typically makes use of dynamic programming algorithms and efficient search.

4.3.2. Features for Log-Linear Models

An important consideration is the derivation of features for the translation system. The features are complex enough that they model the translation process accurately and provide useful information, but also require that they can be computed efficiently and can be handled by the algorithms used for decoding.

One feature common to all translation systems is the output language model log P(e). It is not common to use more than one language model feature, with a relatively simple language model being used during a first pass of decoding and a more accurate, but a computationally more expensive, language model being used for rescoring a lattice of hypotheses to refine the output [07Bra]. The features depend on any property of the two sentences, and features can have varying complexity. Features from generative model P(e |f), P(f |e) in either translation direction can be used. Other, simpler features used include: sentence length distributions that condition the length of the output sentence on that of the input sentence; distortion models that consider the reordering of words and phrases; features which reward word pairs found in conventional bilingual dictionaries and lexical features which reward word pairs found in training data. A word insertion penalty is effective in controlling output length and makes a large difference to translation quality [09Koe].

4.4. Model Training

Once model P_Λ (e |f) is defined by deciding on features, the model parameters are needed to estimate from training data. This is typically done in two steps. The first step estimates parameters for the features that make up the model from a large corpus of training data. Most features are estimated on a corpus of parallel text $\{(e_s, f_s)\}_{s=1}^{S}$, though language models and other features for which h(f , e) = h(e), are estimated on the output language only.

The second step is the estimation of the feature weights Λ of the log-linear model. Discriminative training is performed on a development set smaller than the parallel corpus, $\{(e_s, f_s)\}_{s=1}^{\acute{S}}$, and usually translates the source text to minimise the error (under some loss function) compared to a reference translation [02Och].

The models can be used to translate many different types of data, from newswire to web data, to the output from a speech recognition system run on television broadcasts, with each type having different properties desired from its output. Discriminative training can be used to help with domain adaptation, for example to give greater importance to a language model corresponding to the desired domain and modify the insertion penalty. The development set is matched to the text the model will be used to translate, and the parameters are tuned to optimize the system's performance on that type of data.

4.5. Discriminative Training of Log-Linear Models

These discriminative models are trained by optimizing the parameters Λ to maximize some objective function. There are many possibilities for the objective function; the challenge is to find an objective function that is feasible to compute and improves translation quality. First attempts at performing discriminative training optimized the feature weights to maximize the likelihood of the entire parallel text:

$$\hat{\Lambda}=\operatorname{argmax}_{\Lambda}\sum_{s=1}^{S}P_{\Lambda}(e_s|f_s) \tag{4.5}$$

The maximum entropy training is known as equation 4.5 since the maximum likelihood estimate of Λ produces the model which has maximum entropy on the training data consistent with the constraints placed on the model. The number of features is limited since they add features one by one according to which one gives the largest increase in likelihood.

However, calculation of the denominator is computationally intensive, since it requires a sum over all possible translations e, and many iterations of training are required. Since some of the sentences may have more than one reference translation, a modified training objective is introduced. This assigns an equal weight to each reference sentence where there are multiple references. Some reference sentences may not appear in the N-best list, so the reference for training purposes is the sentence from the N-best list which contains the minimal number of word errors relative to the reference translations [07Liu]. Optimization of the likelihood of training data does not necessarily guarantee improved translation quality on unseen text, which is the ultimate aim. There is a range of evaluation metrics for machine translation output, which discriminative training can be used; the metric can be picked with which the system is to be evaluated and optimized according to this metric during discriminative training.

4.6. Alignment Modeling

In Automatic Speech Recognition (ASR), a transcription e is estimated from acoustic data O using the following equation [00Jur]:

$\hat{e} = \text{argmax}_e\ P(O|e)P(e)$ (4.6)

This can be viewed as a source-channel model, with the source word sequence e, modeled by the language model P(e), being passed through a noisy channel to produce O: this task is to determine e given only the information contained in O; complex statistical models are used whose parameters are estimated from training data to ensure they accurately represent that data.

The translation model P(f |e) can be viewed as an analogue of the acoustic model P(O|e). In machine translation, however, there is an additional level of complexity that does not occur in ASR. Whereas the words in the ASR transcription e occur in the same order in which they are uttered, and therefore the same order in which their representations appear in O, the ordering of words in the input and output languages need not to be the same. It is necessary to find the alignment of words in order to find their translations. Alignment is considered at three levels: document (determining which input language document is a translation of which output language document); sentence (determining a correspondence between the sentences within the document); and word/phrase level. The problem of determining alignment at the sentence level is given equivalent documents, but it concentrate on determining word and phrase level alignments once sentence pairs have been matched.

A generative model of alignment and view f are being generated from e. For alignment, the source language is the language generated from and the target language. The alignment models can be used in both translation directions, i.e. the input language sentence can be viewed as being generated from the output language sentence, and vice versa. Given a source language sentence e and target language sentence f, we introduce a hidden alignment variable A which determines the correspondence between the words and phrases in the two sentences, and calculate the translation probability as the following equation:

$P(f\ |e) = \sum_e P(f, A \vee e)$ (4.6)

where the sum is taken over all possible alignments A. The number of possible alignments between e and f is $2^{|e||f\,|}$ and increases exponentially as the sentence lengths increase.

This poses a problem for modeling due to the complexity of the models involved. The area of alignment modeling is concerned with developing techniques to overcome these problems and work out the translational correspondence between sentences.

4.7. Language Modeling

An important part of the translation system is the modeling of the output language, since it allows us to favor directly translation hypotheses that are grammatical sentences. The sentence is generated left to right and that each word depends on the previous words, i.e. the probability of sentence $e = e_1, e_2, \dots, e_I = e_1^I$ is given by

$$P(e_1^I) = \prod_{i=1}^{I} P(\ e_i \ V \ e_1, e_2, \dots, e_{i-1}) \qquad (4.7)$$

For computational reasons, we make the approximation that each word depends only on n−1 previous words; this is known as an n-gram language model [29] The sentence probability is approximated by

$$P(e_1^I) \approx \prod_{i=1}^{I} P\ (e_i \ V e_{i-n+1}, \dots, e_{i-1}) \qquad (4.8)$$

and the n-gram probabilities $P(e_i|e_{i-n+1}^{i-1})$ are estimated from large amounts of monolingual data in the output language. The maximum likelihood estimate of $P(e_i|e_{i-n+1}^{i-1})$ is

$$P(e_i|e_{i-n+1}) = \frac{c(e_{i-n+1}^{i})}{c(e_{i-n+1}^{i-1})} \qquad (4.9)$$

where c are the counts of the word sequences in training data. These maximum likelihood estimates can suffer from data sparsity, i.e. they are inaccurate when there are few examples of an n-gram in the training data. They also assign zero probability mass to n-grams or words that do not occur in training data.

Various smoothing methods are employed to adjust the maximum likelihood estimates to produce more accurate probability distributions [99Che]. The interpolation of probabilities can be used where a lower order distribution is interpolated with a higher order distribution; backoff, is the use of a lower order n-gram probability when a higher order n-gram is not available; discounting subtracts from non-zero counts in order that probability mass can be distributed among n-grams with zero count, usually according to a lower order distribution. The general form of a large number of language models is as equation 4.10:

$$P_{smooth}(e_i|e_{i-n+1}^{i-1}) = \begin{cases} \rho(e_i|e_{i-n+1}^{i-1}) & , \text{ if } c(e_{i-n+1}^{i}) > 0 \\ \gamma(e_i|e_{i-n+1}^{i-1})\, P_{smooth}(e_i|e_{i-n+2}^{i-1}) & , \text{otherwise} \end{cases} \qquad (4.10)$$

for some probability distribution ρ defined over n-grams that occur in the training data, where $\gamma(e_i|e_{i-n+1}^{i-1})$ is the backoff weight and is normalized to ensure a valid probability distribution. All such models require significant computation to ensure probabilities [07Bra] are normalized.

The main benefit of this scheme is that training is efficient due to the fact that normalization of probabilities is not performed. Despite its simplicity, it is competitive with more sophisticated methods, especially as the amount of training data increases, and can be trained on large amounts of data where other models would be too complex.

4.8. Phrase-Based Statistical Machine Translation (PBSMT)

In phrase-based translation, the aim is to reduce the restrictions of word-based translation by translating whole sequences of words, where the lengths may differ. The sequences of words are called blocks or phrases, but typically are not linguistic phrases. It is the main component of the translation system. The hypotheses are generated by concatenating target language phrases. A phrase is simply a contiguous sequence of words. The pairs of source and corresponding target phrases are extracted from the word-aligned bilingual training corpus.

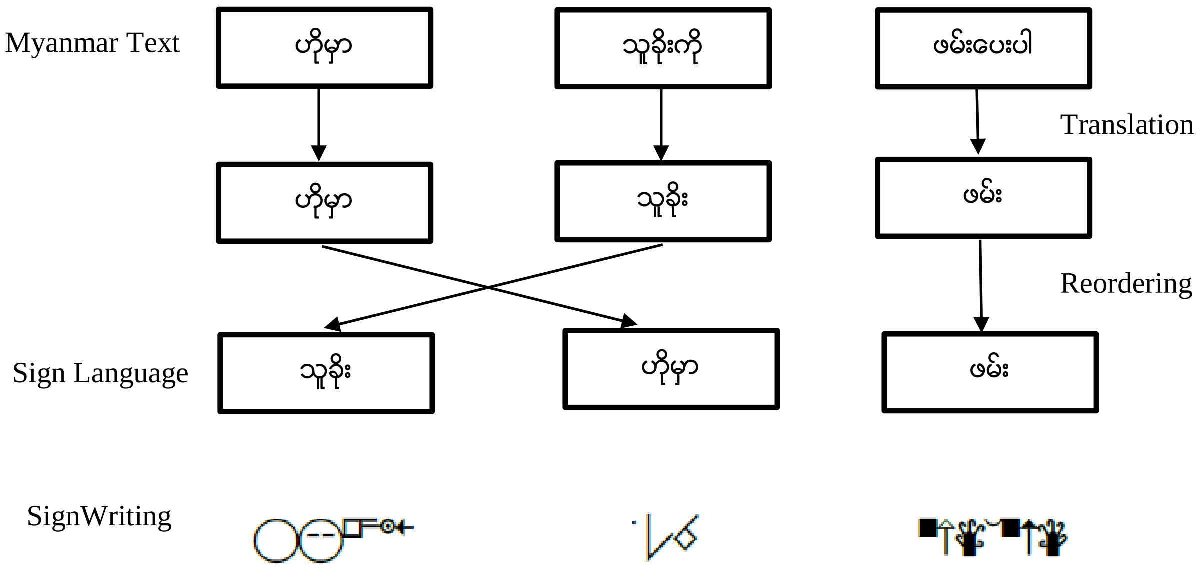


Figure 4.3. Phrase-Based Translation

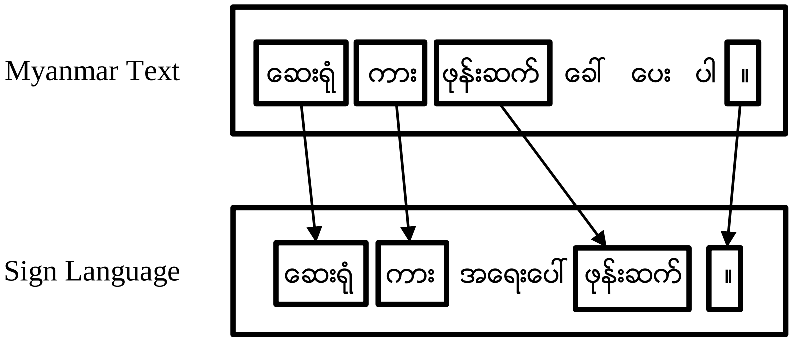
A distortion model is applied to account for changes in phrase order between the input and output languages. The distortion models used during decoding are simple due to the computational expense of allowing all possible permutations of phrases: it has been shown that the search for the best permutation is NP-complete. Some phrase-based systems allow only monotonic alignments, which leads to fast decoding but causes problems for languages with different word orders.

Moses [09Koe] is an open source decoder with the functionality of Pharaoh, but also allows factored translation models where the text itself is augmented with additional information such as part of speech tags or other morphological, syntactic or semantic information. Confusion network decoding allows it to take ambiguous input, such as the output from a speech recognizer.

The distortion model used by Moses allows positioning of a phrase relative to the last word in the previous phrase, independent of the content of the phrases, and sets a distortion limit for the maximum distance a phrase is allowed to move. The cost of a reordering is defined to be the sum of the individual jump sizes; this cost is used as a feature in the log-linear model [11Ber]. Example of phrase-based translation is shown in Fig 4.3.

4.9. Hierarchical Phrase-Based Statistical Machine Translation (HPBSMT)

The hierarchical phrase-based SMT (HPBSMT) approach [30] is a model based on synchronous context-free grammar. The model is able to be learned from a corpus of unannotated parallel text. The advantage of this technique offers over the phrase-based approach is that the hierarchical structure is able to represent the word reordering process.



X → X_1 X_2 X_3ခေါ် ပေး ပါ X_4 | X_1 X_2 အရေးပေါ် X_3 X_4

Figure 4.4. Hierarchical Phrase-Based Translation

The reordering is represented explicitly rather than encoded into a lexicalized reordering model (commonly used in purely phrase-based approaches). This makes the approach particularly applicable to language pairs that require long-distance reordering during the translation process.

4.10. Operation Sequence Model (OSM)

The Operation Sequence Model explains the translation process as a linear sequence of operations which generate a source and target sentence in parallel. Possible operations are (i) generation of a sequence of source and/or target words (ii) insertion of gaps as explicit target positions for reordering operations, and (iii) forward and backward jump operations which do the actual reordering. The probability of a sequence of operations is defined according to an n-gram model, i.e., the probability of an operation depends on the n-1 preceding operations. Since the translation (generation) and reordering operations are coupled in a single generative story, the reordering decisions may depend on preceding translation decisions, and translation decisions may depend on preceding reordering decisions. This provides a natural reordering mechanism which is able to deal with local and long distance reordering consistently [14Nad].

The generative story of the model is motivated by the complex reordering in the Myanmar Text to Myanmar SignWriting task. The SignWriting symbols are generated in linear order, while the Myanmar texts are generated in parallel with SignWriting translations. Mostly the generation is done monotonically. Occasionally the translator inserts a gap on the Myanmar texts side to skip some words to be generated later. Each inserted gap acts as a designated landing site for the translator to jump back to. When the translator needs to cover the skipped words, it jumps back to one of the open gaps. After this is done, the translator jumps forward again and continues the translation. OSM model uses five translation and three reordering operations which are repeatedly applied in a sequence. Five translation operations are as follow:

- Generate (X,Y),
- Continue Source Cept,
- Generate Identical,
- Generate Source only (X), and

- Generate Target only (Y).

Three reordering operations are as follow:

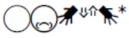
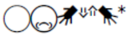
- Insert Gap,
- Jump Back (N), and
- Jump Forward.

4.10.1. Five Translation Operations

The following is a definition of each of these five translation operations [17Phi].

- Generate (X, Y): X and Y are Myanmar texts and Myanmar SignWriting cepts respectively, each with one or more words. Words in X (Myanmar text) may be consecutive or discontinuous, but the words in Y (SignWriting) must be consecutive. This operation causes the words in Y and the first word in X to be added to the SignWriting and Myanmar text strings respectively, that were generated so far. Subsequent words in X are added to a queue to be generated later. All the SignWriting words in Y are generated immediately because SignWriting (target-side) is generated in linear order as per the assumption of the model. The generation of the second (and subsequent) Myanmar words in a multi-word cept can be delayed by gaps jumps and other operations defined below.
- Continue Source Cept: The Myanmar words added to the queue by the Generate (X, Y) operation are generated by the Continue Source Cept operation. Each Continue Source Cept operation removes one Myanmar word from the queue and copies it to the Myanmar string. If X contains more than one Myanmar word, say n many, then it requires n translation operations, an initial Generate ($X_1 \ldots X_n$; Y) operation and n-1 Continue Source Cept operations. For example, "မီးသတ် ကား – returned" is generated by the operation Generate (မီးသတ် ကား, returned), which adds "မီးသတ်" and "returned" to the SignWriting and Myanmar text strings and "ကား" to a queue. A Continue Source Cept operation later removes "ကား" from the queue and adds it to the Myanmar string.
- Generate Source Only (X): The words in X are added at the current position in the Myanmar string. This operation is used to generate a Myanmar word with no corresponding SignWriting. It is performed immediately after its preceding

Myanmar word is covered. This is because there is no evidence on the SignWriting-side which indicates when to generate X. Generate Source Only (X) helps us learn a source word deletion model. It is used during decoding, where a Myanmar word X is either translated to some SignWriting by a Generate (X, Y) operation or deleted with a Generate Source Only (X) operation.

- Generate Target Only (Y): The words in Y are added at the current position in the SignWriting string. This operation is used to generate SignWriting word with no corresponding Myanmar word. This operation is not utilized in MTU-based decoding where it is hard to predict when to add unaligned target words, during decoding. The alignments are modified to remove this, by aligning unaligned target words. In phrase-based decoding, however, this is not necessary, as it can easily predict unaligned target words where they are present in a phrase pair.
- Generate Identical: The same word is added at the current position in both the Myanmar and SignWriting strings. The Generate Identical operation is used during decoding for the translation of unknown words. The probability of this operation is estimated from singleton Myanmar words that are translated to an identical string. For example, for a tuple "cpr – ", where Myanmar "cpr" was observed exactly once during training, a Generate Identical operation is used rather than Generate (cpr, ).

4.10.2. Reordering Operations

This section now discusses the set of reordering operations used by the generative story. Reordering has to be performed whenever the Myanmar word to be generated next does not immediately follow the previously generated Myanmar word. During the generation process, the translator maintains an index which specifies the position after the previously covered Myanmar word (j), an index (Z) which specifies the index after the right-most Myanmar word covered so far, and an index of the next Myanmar word to be covered (j_0).

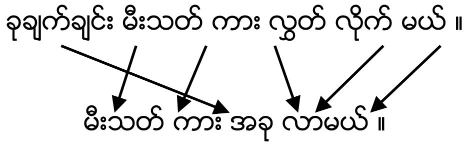


Figure 4.5. Example of Reordering Operation

The set of reordering operations used in generation depends upon these indexes, as shown in Fig 4.5 [14Nad].

- Insert Gap: This operation inserts a gap which acts as a placeholder for the skipped words. There can be more than one open gap at a time.
- Jump Back (W): This operation lets the translator jump back to an open gap. It takes a parameter W specifying which gap to jump to. The Jump Back (1) operation jumps to the closest gap to Z, Jump Back (2) jumps to the second closest gap to Z, etc. After the backward jump the target gap is closed.
- Jump Forward: This operation makes the translator jump to Z. It is performed when the next word to be generated is to the right of the last word generated and does not follow it immediately. It will be followed by an Insert Gap or Jump Back (W) operation if the next source word is not at position Z.

4.10.3. Conversion Algorithm

An aligned bilingual sentence pair is converted to a sequence of operations. This model is estimated from a sequence of operations obtained through the transformation of a word-aligned bilingual corpus. An operation can be to generate source and target words or to perform reordering by inserting gaps and jumping forward and backward. Let $O = o_1, \ldots, o_J$ be a sequence of operations as hypothesized by the translator to generate a word-aligned bilingual sentence pair < F,E,A >; the translation model is then defined as equation 4.11:

$$P_T\,(F,E,A) = p(o_1, \ldots, o_J\,) = \prod_{j=1}^{J} p(o_j \mid o_{j-n+1} \ldots o_{j-1}) \qquad (4.11)$$

where n indicates the amount of context used, A defines the word-alignment function between E and F. The translation model is implemented as an N-gram model of operations using the KENLM toolkit. The translate operations in our model (the operations with a name starting with Generate) encapsulate tuples. Tuples are minimal translation units extracted from the word-aligned corpus. The idea is similar to N-gram-based SMT except that the tuples in the N-gram model are generated monotonically. The restriction of monotonicity in the model is not imposed but integrates reordering operations inside the generative model.

Like in the tuple N-gram model, there is a one-to-one correspondence between aligned sentence pairs and operation sequences, i.e., we get exactly one operation sequence per bilingual sentence given its alignments. The corpus conversion algorithm maps each bilingual sentence pair given its alignment into a unique sequence of operations deterministically; it thus maintains a one-to-one correspondence. This property of the model is useful as it addresses the spurious phrasal segmentation problem in phrase-based models. A phrase-based model assigns different scores to a derivation based on which phrasal segmentation is chosen. Unlike this, the OSM model assigns only one score because the model does not suffer from spurious ambiguity.

Corpus Conversion Algorithm

Input	Output
$E_1 \dots E_n$ SignWriting Cepts	$\langle O \rangle = o_1 \dots o_n$ Vector of Operations
$F_1 \dots F_n$ Myanmar Cepts	
$a_1 \dots a_n$ Alignment between E and F	
i Position of current SignWriting cept	F_{ai} Sequence of Myanmar words linked to E_i
j Position of current Myanmar word	$\lvert F_{ai} \rvert$ # of Myanmar words linked with E_i
j' Position of next Myanmar word	k # of already generated Myanmar words for E_i
N Total number of SignWriting cepts	a_{ik} Position of k^{th} Myanmar translation of Ei
f_j Myanmar word at position j	Z Position after right-most generated Myanmar word
E_i SignWriting cept at position i	
	S(W) Position of the first word of a target gap W

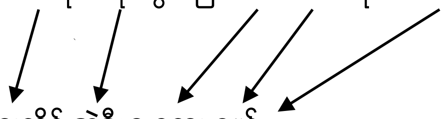
```
i:= 0; j := 0; k := 0
while $f_j$ is an unaligned word do
    O.push(Generate Source Only ($f_j$ ))
    j := j + 1
while $E_i$ is an unaligned cept do
    O.push(Generate Target Only ($E_i$))
    i := i + 1
Z := j
while i < N do
    j' := $a_{ik}$
    if j < j' then
        if $f_j$ was not generated yet then
            O.push(Insert Gap)
        if j = Z then
            j := j'
        else
            O.push(Jump Forward)
            j := Z
    if j' < j then
        if j < Z and $f_j$ was not generated yet then
            O.push(Insert Gap)
```

```
                  W := relative position of target gap (j)
                  O.push(Jump Back (W))
                  j := S(W)
        if j < j' then
                  O.push(Insert Gap)
                  j := j'
        if k = 0 then
                  O.push(Generate ($F_{ai}$ , $E_i$)) {or Generate Identical}
        else
                  O.push(Continue Source Cept)
        j := j + 1; k := k + 1
        while $f_j$ is an unaligned word do
                  O.push(Generate Source Only ($f_j$ ))
                  j := j + 1
        if Z < j then
                  Z := j
        if k = |$F_{ai}$| then
                  i := i + 1; k := 0
        while $E_i$ is an unaligned word do
                  O.push(Generate Target Only ($E_i$))
                  i := i + 1
return 0
```

Figure 4.6. Corpus Conversion Algorithm for OSM Model

The following example shows step by step by means of an example how the conversion is done. The values of the index variables are displayed at each point in table 4.1.

Myanmar text: ဆေးဆိုင် ကို သွား ပြီး ဆေး ဝယ် လိုက် ပါ ။



SignWriting: ဆေးဆိုင် အဲ့ဒီမှာ ဆေး ဝယ် ။

4.10.4. Discriminative Model

A log-linear approach [33] is used to make use of standard features along with several novel features that introduce to improve end-to-end accuracy. It searches for a target string E which maximizes a linear combination of feature functions in equation 4.12:

$$\hat{E} = \arg \max_{E} \sum_{j=1}^{J} \lambda_j h_j (F, E)\} \qquad (4.12)$$

where λ_j is the weight associated with the feature h_j(F,E).

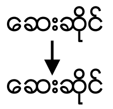
Apart from the OSM model and standard features such as target-side language model, length bonus, distortion limit and IBM lexical features [03Koe] are used:

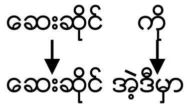
- Deletion Penalty: Deleting a source word (Generate Source Only (X)) is a common operation in the generative story. Because there is no corresponding target-side word, the monolingual language model score tends to favor this operation. The deletion penalty counts the number of deleted source words.
- Gap and Open Gap Count: These features are introduced to guide the reordering decisions. A large amount of reordering will be observed in the automatically word aligned training text. However, given only the source sentence (and little world knowledge), it is not realistic to try to model the reasons for all of this reordering. Therefore, a more robust model can be used that reorders less than humans do. The gap counts feature sums to the total number of gaps inserted while producing a target sentence.

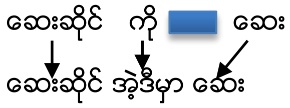
Table 4.1. Step-Wise Generation of Myanmar Written Text to Myanmar Sign Language

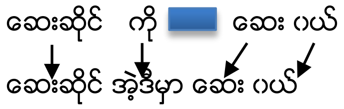
Operations	Generation	State

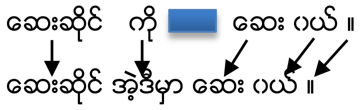
		i=0, j=0, k=0
		Z=0, j'=0

Generate(ဆေးဆိုင်,ဆေးဆိုင်)		i=1, j=1, k=0
		Z=1, j'=1

Generate(ကို, အဲ့ဒီမှာ),
Generate Source Only(သွား)
		i=2, j=3, k=0
		Z=3, j'=4

Insert Gap,
Generate(ဆေး, ဆေး)
		i=3, j=5, k=0
		Z=5, j'=5

Generate(ဝယ်, ဝယ်),
Generate Source Only(လိုက်)
		i=4, j=7, k=0
		Z=7, j'=8

Insert Gap,
Generate(။, ။)
		i=5, j=9, k=0
		Z=9

The open gap count feature is a penalty paid once for each translation operation (Generate(X, Y), Generate Identical, Generate Source Only (X)) performed whose value is the number of currently open gaps. This penalty controls how quickly gaps are closed.

- Distance-based Features: Two distance-based features are used to control the reordering decisions. One of the features is the Gap Distance which calculates the distance between the first word of a source cept X and the start of the left-most gap. This cost is paid once for each translation operation (Generate, Generate Identical, Generate Source Only (X)). For a source cept covering the positions $X_1 \dots X_n$, we get the feature value $g_j = X_1$-S, where S is the index of the left-most source word where a gap starts. Another distance-based penalty used in our model is the Source Gap Width. This feature only applies in the case of a discontinuous translation unit and computes the distance between the words of a gappy cept. Let f = f_1 …, f_{i} , … ,f_n be a gappy source cept where x_i is the index of the i^{th} source word in the cept f. The value of the gap width penalty is calculated as equation 4.13:

$$w_j = \sum_{i=2}^{n} x_i - x_{i-1} - 1 \qquad (4.13)$$

In the next section, the evaluation criteria of machine translation will be explained.

4.11. Evaluation for Machine Translation

The quality of output from a translation system is generally judged relative to a reference translation (or reference translations) of the sentence in question. The criteria commonly used or evaluation of machine translation by human evaluators are fluency and adequacy [93Whi]. Fluency is a measure of the quality of language of the hypothesis translation, ranging from a grammatically flawless sentence to incomprehensible. Adequacy is a measure of how much of the meaning of the reference translation is expressed in the hypothesis translation. Human evaluation of machine translation output is expensive in terms of both the time taken and the financial expense of paying bilingual and monolingual evaluators. There are also issues with inter-evaluator agreement (different evaluators may give different results) and intra-evaluator consistency (the same evaluator may produce different results at different times). Therefore, it may not be possible to measure small changes in the quality of the output. When developing machine translation systems, feedback is required as rapidly as possible after a change has been made to the system, in order to determine quickly the success of the approaches applied. We therefore require an automated method of evaluating the output of a system [05Ldc].

Automatic evaluation of machine translation quality is essential to developing high-quality machine translation systems because human evaluation is time consuming, expensive, and irreproducible. If it has a perfect automatic evaluation metric, translation system for the metric can be tuned. In the natural language processing, there are many kinds of automatic evaluation methodology (e.g., BLEU, WER, and RIBES, etc.). In the research, two automatic criteria will mainly be used for the evaluation of the machine translation output –BLEU and RIBES.

4.11.1. BiLingual Evaluation Understudy (BLEU)

The BiLingual Evaluation Understudy (BLEU) [02Pap] is a metric widely used for automatic evaluation of machine translation output. The basic premise is that translation of a piece of text is better if it is close to a high-quality translation produced by a professional translator. The translation hypothesis is compared to the reference translation, or multiple reference translations, by counting how many of the n-grams in the hypothesis appear in the reference sentence(s); better translations will have a larger number of matches. The ranking of sentences using BLEU score has been found to closely approximate human judgment in assessing translation quality.

A candidate translation that is longer than the reference sentences will have a lower modified n-gram precision since the denominator is larger. However, we also wish to penalize sentences that are shorter than the reference transcriptions. A brevity penalty is calculated over the entire corpus and penalizes the model if the candidate translation sentences are too short compared to the reference translations. This brevity penalty is given by the equation 4.14,

$$b(c,r) = \begin{cases} 1 & ,\text{if } c > r \\ e^{1-\frac{r}{c}} & ,\text{if } c \leq r \end{cases} \tag{4.14}$$

where c = $P_{e\in C}$ |e| is the length of the candidate translation corpus and r is the effective reference corpus length. The effective corpus length can be calculated in a number of ways. The BLEU score of a corpus C is shown in equation 4.15.

$$\text{BLEU} = b(c, r)\ \exp\ \left(\textstyle\sum_{n=1}^{N} \lambda_n \log P_n\ (C)\right) \quad (4.15)$$

where N is the maximum length of n-gram considered and λ_n are positive weights such that $\sum_{n=1}^{N} \lambda_n = 1$. The most commonly used values for these parameters are N = 4, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \frac{1}{4}$. This score ranges from zero to one, with a higher score indicating a better translation, but results are often quoted by scaling to a number between 0 and 100. Very few translations will achieve a score of 100, unless they are identical to one of the reference translations.

4.11.2. Rank-Based Intuitive Bilingual Evaluation Score (RIBES)

The focus of the RIBES metric [10Iso] is word order. It uses rank correlation coefficients based on word order to compare SMT and reference translations. The primary rank correlation coefficients used are Spearman's ρ, which measures the distance of differences in rank (equation 4.16), and Kendall's τ, which measures the direction of differences in rank (equation 4.17).

$$\text{Spearman's } \rho = 1 - \frac{\sum_i d_i^2}{{}_{n+1}C_3} \quad (4.16)$$

$$\text{Kendall's } \tau = 2 * \frac{(\text{The numbers of increasing pairs})}{(\text{numbers of all pairs})} - 1 \quad (4.17)$$

Both ρ and τ have the same range [-1,1]. Therefore, it needs to normalize to get the positive values as equation 4.18 and equation 4.19.

$$\text{Normalized Spearman's } \rho \text{ (NSR)} = (\rho + 1)/2 \quad (4.18)$$

$$\text{Normalized Kendall's } \tau \text{ (NKT)} = (\tau + 1)/2 \quad (4.19)$$

These measures can be combined with precision P and modified to avoid overestimating the correlation of only corresponding words in the SMT and reference translations as equation 4.20:

$$\text{NSR } P\ \alpha \text{ and NKT } P\ \alpha \quad (4.20)$$

where α is a parameter in the range 0 < α < 1. Kendall's τ is usually smaller values than Spearman's ρ.

4.12. Error Analysis

WER (Word Error Rate) [13Mar] is defined as Levenshtein distance between the words of the system output and the words of the reference translation divided by the length of the reference translation. Levenshtein distance is calculated using dynamic programming to find an optimal alignment between the output of machine translation and the reference translation. Each word machine translation output is aligned to one or zero words in reference translation, and vice versa. The case where the reference word is aligned to zero are called the deletion, whereas the alignment of word of machine translation to zero is called insertion. If a reference word matches the MT output word it is aligned to, it is a substitution. WER is the sums of the number of substitutions (S), insertions (I), and deletions (D) divided by the number of words in the reference translation (N) as equation 4.21:

$$\text{WER} = \frac{\text{S+I+D}}{\text{N}} *100 \qquad (4.21)$$

where I is the number of insertion, D is the number of deletions, S is the number of substitutions, and C is the number of Corrected words. Note that if the number of insertions is very high, the WER can be greater than 100%. In this research, we used the SCLITE (score speech recognition system output) program from the NIST scoring toolkit SCTK version 2.4.10 for making dynamic programming-based alignments between reference (ref) and hypothesis (hyp) and calculation of WER.

4.13. Summary

This chapter has introduced the framework used for statistical machine translation and examined some of the models used. The steps in the process have been described, including sources of data and its pre-processing, building of alignment models using parallel text, log-linear models of translation, features for log-linear models, discriminative training of models, the decoding process itself as well as automated metrics fore valuation of translation quality and error analysis.

CHAPTER 5
SYSTEM IMPLEMENTATION

This chapter will describe the detail system implementation of this research. It will contain system design, data collection, segmentation and step-by-step process of this research.

5.1. System Design

This research mainly proposes the three statistical machine translation approaches to translate between MWT-MSW, MWT-MSL and MSL-MSW, in both directions. The basic need of this research is MWT, MSL, MSW parallel corpus.

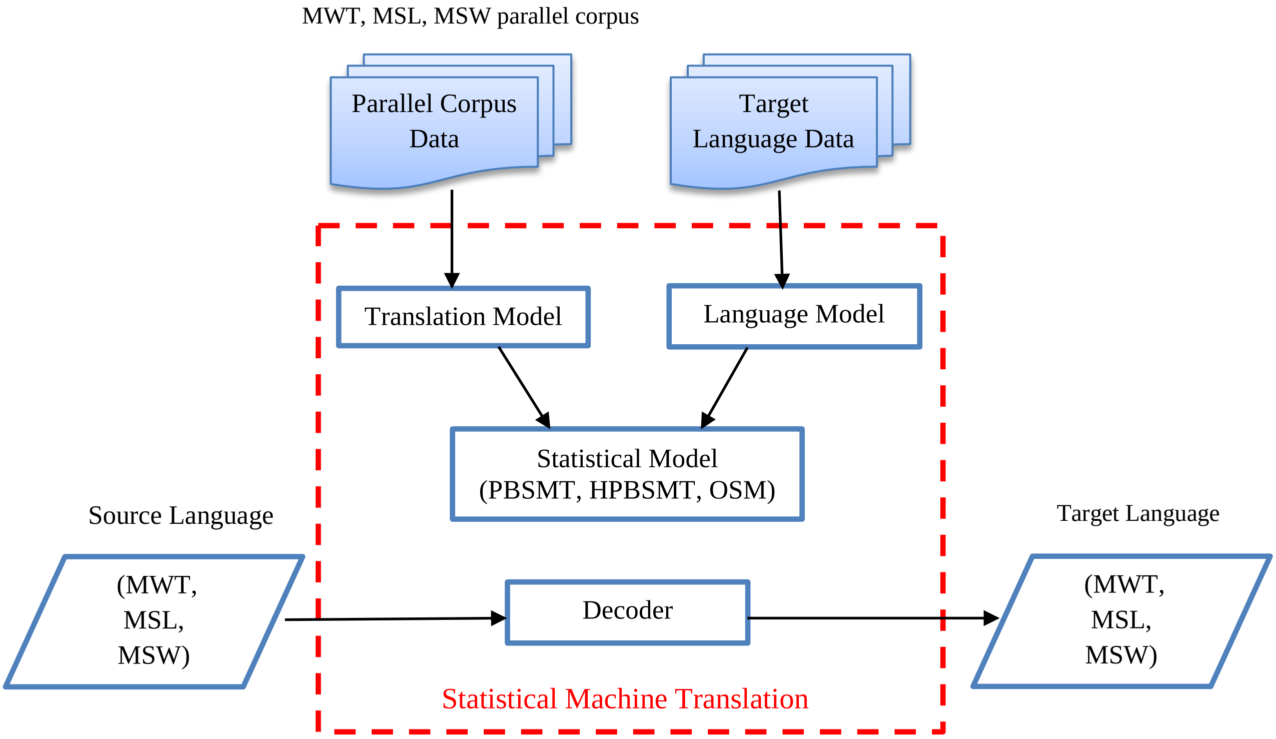


Figure 5.1. Overview of System Design

The system design of this research is shown in Fig 5.1. Firstly, the bilingual data is needed to prepare. Then, this bilingual parallel data is used to build translation model and the monolingual data is used for language model. Then, the statistical approaches are combined with these two models. Decoder uses the decoding algorithm to translate source language to target language. The detail processes of the SMT are shown in Fig 5.2.

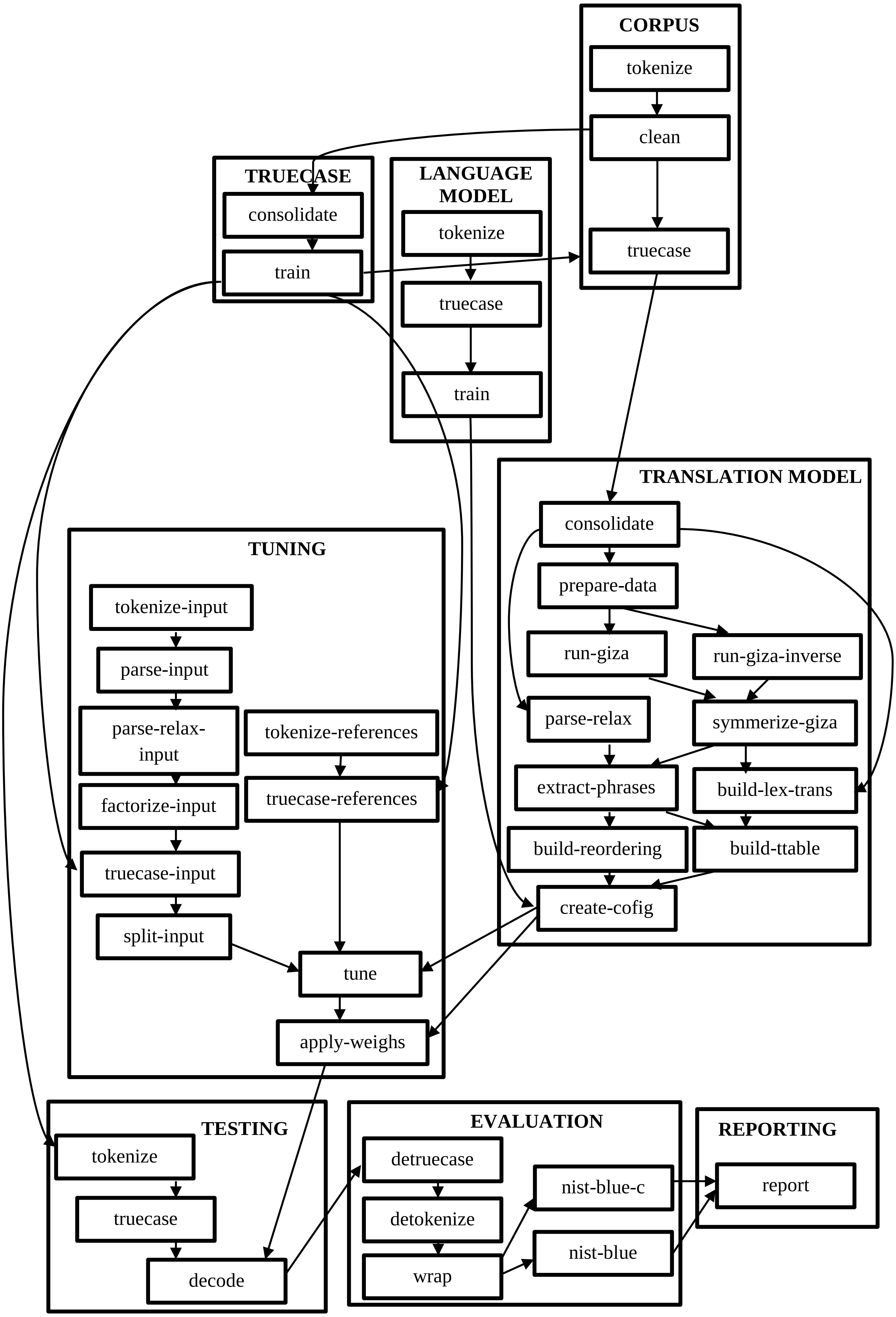


Figure 5.2. Detail Processes of SMT

There are many steps in running SMT: the preparation of training data, building language and translation models, tuning, testing, scoring and analysis of the results. The step-by-step running processes will explain in later sections.

5.2. Corpus Building

The major problem of doing this research is lack of data collection related with Myanmar written text, the transcribed Myanmar sign language and Myanmar SignWriting. Therefore, this may lead to try hard for this problem. This research is mainly focused on the emergency domain and therefore the video data are collected related with fires, earthquake, floods, storms, accidents, police, health, number, date and time, etc. It contains about 888 parallel sentences for Myanmar written text, Myanmar sign language and Myanmar SignWriting. In data preparation, process performs three main processes:

- Video Data Collection,
- Segmentation for MWT and MSL, and
- Manually annotation with Myanmar SignWriting.

The detailed explanations are described in next sections.

5.2.1. Video Data Collection

The spoken style sentences and written style sentences are manually selected from the pamphlets, emergency alerts, emergency books and daily basic conversation book. First of all, sign language trainers are shown for each sentence whether the usage of each sentence is possible or not for Deaf and whether it is used for them. If possible, they discuss with other Deaf or hard-of-hearing people how to translate and show with sign language. It needs to note for each sign and write down with Myanmar text. With this process, the video data are collected –one sentence for one video file.

Figure 5.3. Images for Video Data Collection

Video data are collected with professional Myanmar deaf signers from School for the Deaf (Mandalay), Mary Chapman School for the Deaf Children (Yangon), School for the Deaf, Tarmwe (Yangon) and Myanmar Deaf Society (MDS). There are 13 native signers and sign language trainers involved – four are male and nine are female singers respectively. The participants' ages ranged from 20 to 48 years.

These recording steps made sure that the translated sign sentences are fully independent from the Myanmar written sentences. 888 MSL emergency sentences were recorded using a Cannon digital camera and were stored in MPEG-4 format. They were edited using the Cyberlink PowerDirector video editing program. The size of the recorded video frame is 1280 x 720 pixels.

5.2.2. Segmentation

In SMT, word segmentation is a necessary step in order to yield a set of tokens upon which the alignment and indeed the whole machine learning process can operate. In Myanmar text, words composed of single or multiple syllables are usually not separated by white space. Spaces are used for easier reading and generally put between phrases, but there are no clear rules for using spaces in Myanmar language. Based on the previous studies relating to effectiveness of Myanmar word segmentation schemes for SMT [13Yek], three segmentation schemes are used for MWT and MSL.

- Syllable segmentation for MWT and MSL,
- Word segmentation for MWT and
- Sign Unit based segmented for MSL.

5.2.2.1. Syllable Segmentation for MWT and MSL

Syllable segmentation is an important preprocess for many natural language processing (NLP) such as Romanization, transliteration and grapheme-to-phoneme (g2p) conversion. It is a syllable segmentation tool for Myanmar language (Burmese) text encoded with Unicode (e.g. Myanmar3, Padauk. Generally, Myanmar words are composed of multiple syllables and most of the syllables are composed of more than one character. Syllables are also basic units for pronunciation of Myanmar words [17Yek]. Focusing on consonant based syllables, the structure of the syllable can be described with Backus Normal Form (BNF) as following equation 5.1:

$$\text{Syllable: } = C\{M\}\{V\}[CK][D] \quad (5.1)$$

where C for consonants, M for medials, V for vowels, K for vowel killer character and D for diacritic characters [17Mya]. Myanmar syllable segmentation can be done with rule based [08Zin], finite state automaton (FSA) [12Tin] or regular expression (RE). In the experiments, RE based Myanmar syllable segmentation tool is used, called “sylbreak” [17Yek].

5.2.2.2. Sign Unit based segmentation for MSL

There are different segmentation schemes for Myanmar language sentence and MSL sentence. For MSL sentence, segmentation is based on meaningful MSL word other sign languages such as ASL, BSL and Japanese Sign language (JSL). Some examples of Myanmar sign language word category are repeated sign (e.g. two or more repeated “thank you” sign for “please”), sign with multiple meanings (e.g. one MSL sign for “blood” and “red”), compound sign (e.g. combination of MSL signs “car”, “emergency” and “fire extinguishing” for “fire truck”), name sign (e.g. Pyin Oo Win city), fingerspelling sign (e.g. “O” sign + “2” sign for “O2”), fingerspelling shortcut sign (“O” for Octane, Myanmar consonant “မ ” (Ma) for Mandalay city) and phrase or sentence level signs (e.g. MSL sign for စိတ်ငြိမ်ငြိမ်ထား (calm down), ကားတိုက် (car accident)).

5.2.2.3. Word Segmentation for MWT

In Myanmar written text, spaces are used for separating phrases for easier reading. There are no clear rules for using spaces in Myanmar language, and thus spaces may (or may not) be inserted between words, phrases and even between a root words and their affixes. In this research, manual word segmentation is used for Myanmar written text of parallel corpus. The word segmentation rules are proposed by [15Win].

5.2.3. Manual Annotation with Myanmar SignWriting

After video data collection (described in section 5.2.1), the defining of SignWriting symbols for each sign of MSL had been made. In details, the recorded videos are watched several times for defining both manual and non-manual signs.

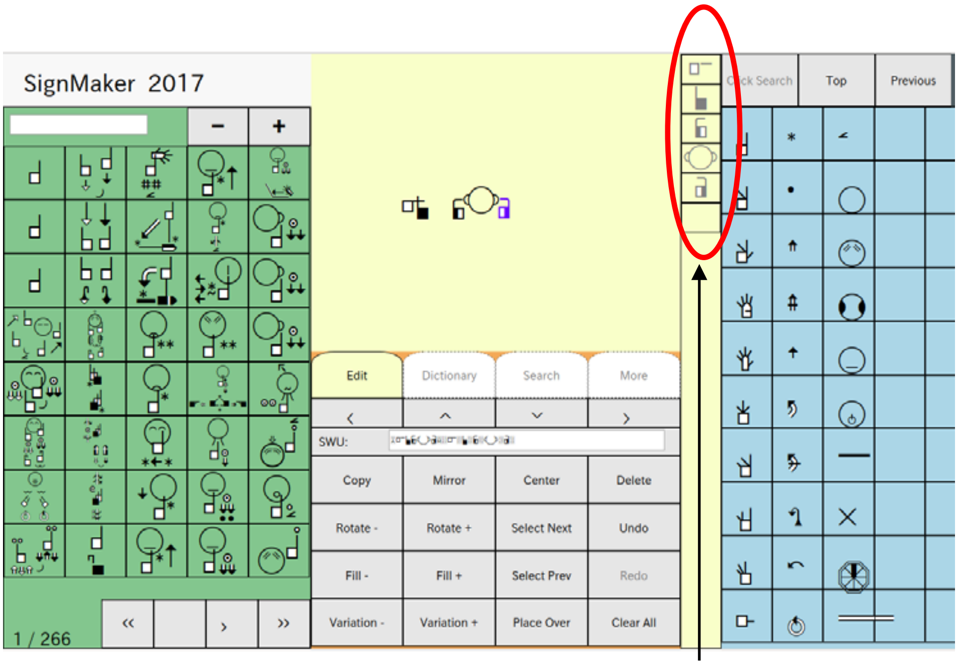
After that, sign symbols are placed on the canvas of SignMaker [15Sig] to form the shape and movement of signs. SignWriting symbols are needed to arrange in a unique sequence. The sign symbols arrangement in SignMaker is shown in Fig 5.4.

There are two ways to prepare SignWriting data: one is the formal SignWriting–based on two-dimensional mathematics and written as a string of ASCII characters and another is Unicode representation of SignWriting symbols. In our work, we use Unicode numbers for SignWriting symbols seeing SignWriting symbols arrangement in SignMaker. An example of SignWriting Unicode character sequences for the MSW word "doctor" is as follows and equivalent MSW can be seen as follows:

English : Doctor

Myanmar : ဆရာဝန်

Unicode Block:**\U1D800\U1DAAA\U1D800\U1DA9C\U1D80A\U1DA9B\U1DAA8 \U1D9FF \U1DA30\U1D80A\U1DA9B**



Sign symbol sequence for Myanmar sign "doctor"

Figure 5.4. An Example of Sign Symbol Sequence Arrangement in SignMaker 2017

5.3. Moses SMT System

This research used the PBSMT, HPBSMT and OSM provided by the Moses toolkit for training the PBSMT, HPBSMT and OSM statistical machine translation systems. The word-segmented source language was aligned with the word-segmented target languages using GIZA++ [00Och]. The alignment was symmetrized by grow-diag-final-and heuristic [03Koe]. The lexicalized reordering model was trained with the msd-bidirectional-fe option [04Til]. KenLM is used for training the five-gram language model with interpolated modified Kneser-Ney discounting [11Hea] [96Che]. Minimum error rate training (MERT) [10Iso] was used to tune the decoder parameters and the decoding was done using the Moses decoder (version 2.1.1). The default setting of Moses is used for all experiments.

5.4. Training

First of all, Moses SMT system is needed to set up. For data preparation, parallel data is needed to train the translation system (i.e., text translated into two different languages) which is aligned at the sentence level.

5.4.1. Data Preparation

To prepare the data for training the translation system, it needs to perform the following steps:

- tokenization: This means that spaces have to be inserted between (e.g.) words and punctuation.
- truecasing: The initial words in each sentence are converted to their most probable casing. This helps reduce data sparsity.
- cleaning: Long sentences and empty sentences are removed as they can cause problems with the training pipeline, and obviously misaligned sentences are removed.

The tokenization can be run as follows:

```
~/mosesdecoder/scripts/tokenizer/tokenizer.perl -l my \
< ~/experiment/msl/my \
> ~/experiment/msl/msw.tok.my
~/mosesdecoder/scripts/tokenizer/tokenizer.perl -l sw \
< ~/experiment/msl/sw \
> ~/experiment/ msl/msw.tok.sw
```

The truecaser first requires training, in order to extract some statistics about the text:

```
~/mosesdecoder/scripts/recaser/train-truecaser.perl \
--model ~/experiment/msl/truecase-model.my --corpus \
~/experiment/msl/msw.tok.my
~/mosesdecoder/scripts/recaser/train-truecaser.perl \
--model ~/experiment/msl/truecase-model.sw --corpus \
~/experiment/msl/ msw.tok.sw
```

Truecasing uses another script from the Moses distribution:

```
~/mosesdecoder/scripts/recaser/truecase.perl \
--model ~/ experiment/msl /truecase-model.my \
< ~/experiment/msl/ msw.tok.my \
> ~/experiment/msl/msw.true.my
~/mosesdecoder/scripts/recaser/truecase.perl \
--model ~/ experiment/msl/truecase-model.sw \
< ~/experiment/msl/ msw.tok.sw \
> ~/experiment/msl/msw.true.sw
```

The script clean-corpus-n.perl is small script that cleans up a parallel corpus, so it works well with the training script. It performs the following steps:

- removes empty lines
- removes redundant space characters
- drops lines (and their corresponding lines), that are empty, too short, too long or violate the 9-1 sentence ratio limit of GIZA++.

```
~/mosesdecoder/scripts/training/clean-corpus-n.perl \
~/experiment/msl/msw.true my sw \
~/ experiment/msl/msw.true.clean 1 80
```

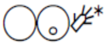
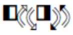
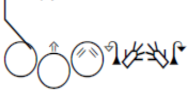
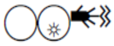
For example: clean-corpus-n.perl raw my sw clean 1 80 takes the corpus files raw.my and raw.sw, deletes lines longer than 80, and creates the output files msw.true.clean.my and msw.true.clean.sw.

Two vocabulary files are generated and the parallel corpus is converted into a numbered format. The vocabulary files contain words, integer word identifiers and word count information, as shown in Fig 5.5.

The sentence-aligned corpus now looks like this:

```
> head -9  corpus/my-sw-int-train.snt
1
28 20 79 356 2
35 63 8 76 5 12 2
1
348 2
43 3 2
1
155 4 230 789 24 89 4 556 219 19 39 2
97 11 44 20 7 848 131 26 68 122 3 2
```

==>corpus/sw.vcb<==

1	UNK	0
2		598
3		98
4		73
5		45
6		45
7		40
8		39
9		35
10		32

==>corpus/my.vcb<==

1	UNK	0
2	။	598
3	တယ်	165
4	ပါ	113
5	မှာ	90
6	မ	89
7	ကို	87
8	နဲ့	78
9	နေ	68
10	ရ	67

Figure 5.5. Vocabulary Files Generated from Parallel Corpus

A sentence pair now consists of three lines: First the frequency of this sentence. In training process this is always one. This number can be used for weighting different parts of the training corpus differently. The two lines below contain word ids of the SignWriting and the Myanmar sentence.

GIZA++ also requires words to be placed into word classes. This is done automatically by calling the mkcls program. Word classes are only used for the IBM reordering model in GIZA++. A peek into the word class file:

```
> head corpus/my.vcb.classes
92              51
95              28
Mask            14
cctv            43
cpr             30
က               41
ကင်း             30
ကစား            31
ကပ်             31
ကမ်း             9
```

5.4.2. Building Alignment Model

GIZA++ [00Och] is a freely available implementation of the IBM models. We need it as a initial step to establish word alignments. The word alignments are taken from the intersection of bidirectional runs of GIZA++ plus some additional alignment points from the union of the two runs.

Running GIZA++ is the most time consuming step in the training process. It also requires a lot of memory (1-2 GB RAM is common for large parallel corpora). GIZA++ learns the translation tables of IBM Model 4, the word alignment file is shown in Fig 5.6.

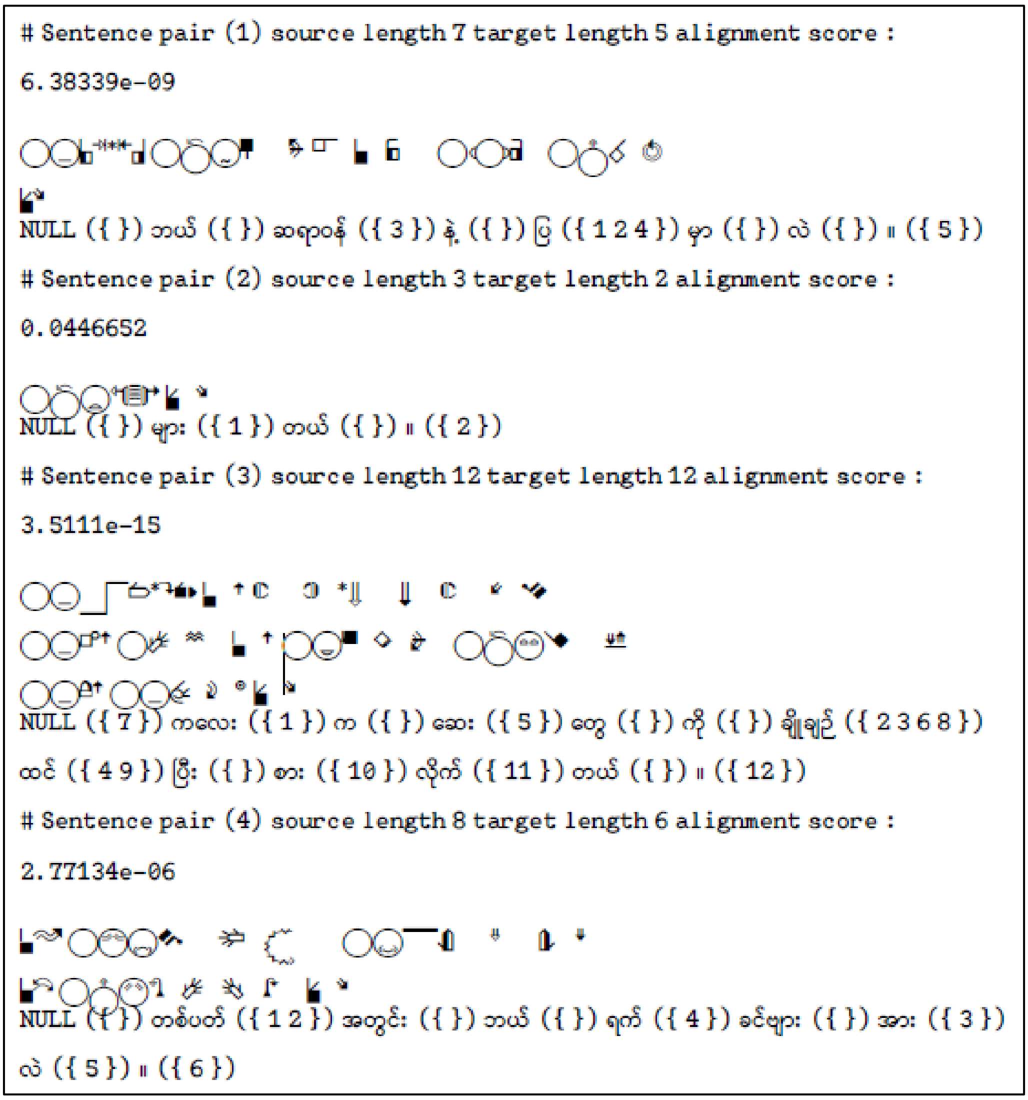
In this file, after some statistical information and the SignWriting sentence, the Myanmar sentence is listed word by word, with references to aligned SignWriting words.

To establish word alignments based on the two GIZA++ alignments, a number of heuristics may be applied. The default heuristic grow-diag-final starts with the intersection of the two alignments and then adds additional alignment points. Other possible alignment methods:

- intersection
- grow (only add block-neighboring points)
- grow-diag (without final step)
- union
- srctotgt (only consider word-to-word alignments from the source-target GIZA++ alignment file)

- tgttosrc (only consider word-to-word alignments from the target-source GIZA++ alignment file)

The alignment metrics are illustrated in the Fig 5.7 below: alignment from P(f | e) and P(e | f).



```
# Sentence pair (1) source length 7 target length 5 alignment score :
6.38339e-09
NULL ({ }) ဘယ် ({ }) ဆရာဝန် ({ 3 }) နဲ့ ({ }) ပြ ({ 1 2 4 }) မှာ ({ }) လဲ ({ }) ။ ({ 5 })
# Sentence pair (2) source length 3 target length 2 alignment score :
0.0446652
NULL ({ }) များ ({ 1 }) တယ် ({ }) ။ ({ 2 })
# Sentence pair (3) source length 12 target length 12 alignment score :
3.5111e-15
NULL ({ 7 }) ကလေး ({ 1 }) က ({ }) ဆေး ({ 5 }) တွေ ({ }) ကို ({ }) ချိုချဉ် ({ 2 3 6 8 })
ထင် ({ 4 9 }) ပြီး ({ }) စား ({ 10 }) လိုက် ({ 11 }) တယ် ({ }) ။ ({ 12 })
# Sentence pair (4) source length 8 target length 6 alignment score :
2.77134e-06
NULL ({ }) တစ်ပတ် ({ 1 2 }) အတွင်း ({ }) ဘယ် ({ }) ရက် ({ 4 }) ခင်ဗျား ({ }) အား ({ 3 })
လဲ ({ 5 }) ။ ({ 6 })
```

Figure 5.6. Word Alignment of my-sw data

The third file contains alignment information, one alignment point at a time, in form of the position of the SignWriting and Myanmar word.

```
==> model/aligned.1.grow-diag-final<==
2-0 3-0 3-1 1-2 0-3 4-3 5-3 6-4
0-0 1-0 2-1
0-0 1-3 2-4 4-5 5-7 7-7 6-8 8-9 9-10 10-10 11-11
0-0 0-1 1-1 2-1 4-2 5-2 3-3 6-4 7-5
```

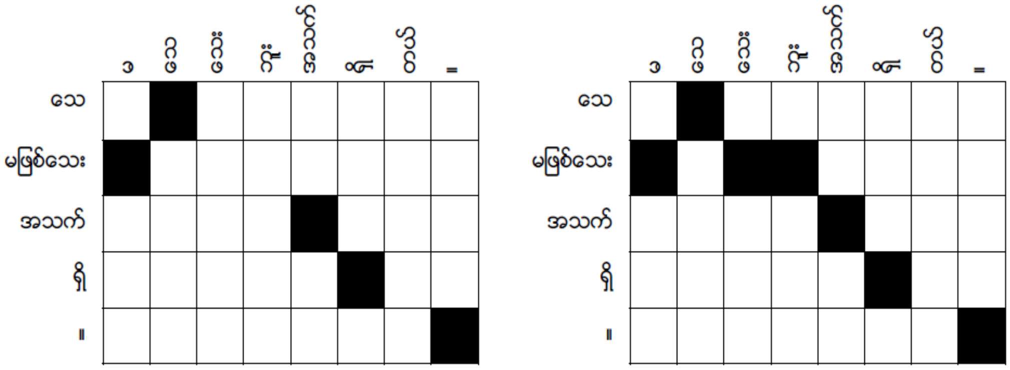
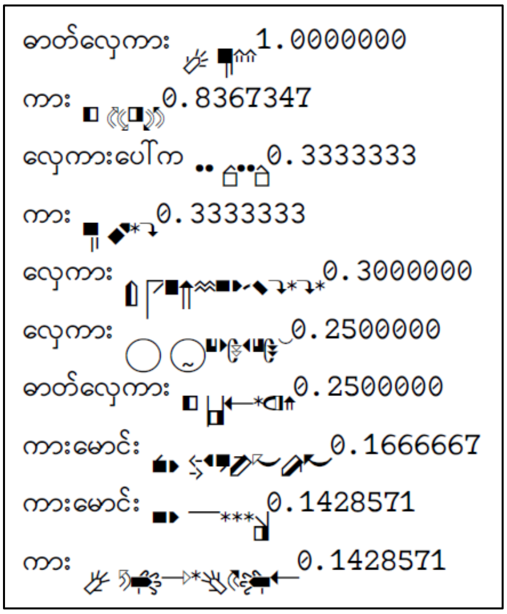


Figure 5.7. (a) Alignment from P(f | e) and (b) Alignment from P(e | f)

5.4.3. Building Translation Table

Given this alignment, it is quite straight-forward to estimate a maximum likelihood lexical translation table. The P(f | e) and the inverse P(e | f) word translation table will be estimated. The top translations for "ကား" ("car" in English) into its relevant SignWriting is shown in Fig 5.8.



ဓာတ်လှေကား 1.0000000
ကား 0.8367347
လှေကားပေါ်က 0.3333333
ကား 0.3333333
လှေကား 0.3000000
လှေကား 0.2500000
ဓာတ်လှေကား 0.2500000
ကားမောင်း 0.1666667
ကားမောင်း 0.1428571
ကား 0.1428571

Figure 5.8. Translations for "ကား" into SignWriting Words

The content of the file in Fig 5.9 is for each line: Myanmar phrase, SignWriting phrase, and alignment points. Alignment points are pairs (Myanmar, SignWriting). Subsequently, a translation table is created from the stored phrase translation pairs. The two steps are separated, because for larger translation models, the phrase translation table does not fit into memory. Fortunately, the phrase translation table never had to store into memory -can construct it on disk.

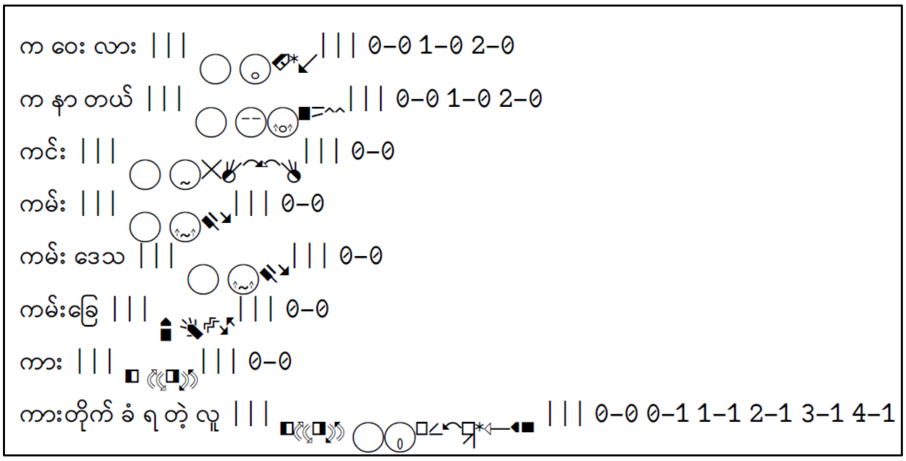

က ဝေး လား ||| ||| 0-0 1-0 2-0
က နာ တယ် ||| ||| 0-0 1-0 2-0
ကင်း ||| ||| 0-0
ကမ်း ||| ||| 0-0
ကမ်း ဒေသ ||| ||| 0-0
ကမ်းခြေ ||| ||| 0-0
ကား ||| ||| 0-0
ကားတိုက် ခံ ရ တဲ့ လူ ||| ||| 0-0 0-1 1-1 2-1 3-1 4-1

Figure 5.9. Alignment between Myanmar and SignWriting Phrases

To estimate the phrase translation probability Ø(e | f) we proceed as follows: First, the extract file is sorted. This ensures that all Myanmar phrase translations for a foreign phrase are next to each other in the file. Thus, we can process the file, one foreign phrase at a time, collect counts and compute Ø(e | f) for that foreign phrase f. To estimate Ø(f | e) the inverted file is sorted, and then Ø(f | e) is estimated for an Myanmar phrase at a time.

Next to phrase translation probability distributions Ø(f | e) and Ø(e | f), additional phrase translation scoring functions can be computed, e.g. lexical weighting, word penalty, phrase penalty, etc. Currently, lexical weighting is added for both directions and a fifth score is the phrase penalty.

Currently, four different phrase translation scores are computed:

1) inverse phrase translation probability Ø(f | e)
2) inverse lexical weighting lex(f | e)
3) direct phrase translation probability Ø(e | f)
4) direct lexical weighting lex(e | f)

5.4.4. Building Reordering Model

By default, only a distance-based reordering model is included in final configuration. This model gives a cost linear to the reordering distance. For instance, skipping over two words costs twice as much as skipping over one word.

The lexicalized reordering models are specified by a configuration string, containing five parts, that account for different aspects:

- Model type - the type of model used
  - wbe - word-based extraction (but phrase-based at decoding). This is the original model in Moses. DEFAULT
  - phrase - phrase-based model
  - hier - hierarchical model
- Orientation - Which classes of orientations that are used in the model
  - mslr - Considers four different orientations: monotone, swap, discontinuous-left, discontinuous-right
  - msd - Considers three different orientations: monotone, swap, discontinuous
  - monotonicity - Considers two different orientations: monotone or non-monotone (swap and discontinuous of the msd model are merged into the non-monotone class)
  - leftright - Considers two different orientations: left or right (the four classes in the mslr model are merged into two classes, swap and discontinuous-left into left and monotone and discontinuous-right into right)
- Directionality - Determines if the orientation should be modeled based on the previous or next phrase, or both.
  - backward - determine orientation with respect to previous phrase DEFAULT
  - forward - determine orientation with respect to following phrase
  - bidirectional - use both backward and forward models
- Language - decides which language to base the model on
  - fe - conditioned on both the source and target languages
  - f - conditioned on the source language only
- Collapsing - determines how to treat the scores
  - allff - treat the scores as individual feature functions DEFAULT
  - collapseff - collapse all scores in one direction into one feature function

As a final step, a configuration file for the decoder is generated with all the correct paths for the generated model and a number of default parameter settings.
This file is called model/moses.ini.

```
#########################
### MOSES CONFIG FILE ###
```

```
#########################
# input factors
[input-factors]
0

# mapping steps
[mapping]
0 T 0

[distortion-limit]
6

# feature functions
[feature]
UnknownWordPenalty
WordPenalty
PhrasePenalty
PhraseDictionaryMemory name=TranslationModel0 num-features=4
path/home/hninwai/experiment/34.my-sw-cross/exp1/baseline/my-sw /model/phrase-
table.1 input-factor=0 output-factor=0
LexicalReordering name=LexicalReordering0 num-features=6 type=wbe-msd-
bidirectional-fe-allff input-factor=0 output-factor=0
path/home/hninwai/experiment/34.my-sw-cross/exp1/baseline/my-sw
/model/reordering-table.1.wbe-msd-bidirectional-fe.gz
Distortion
KENLM name=LM0 factor=0 path/home/hninwai/experiment/34.my-sw-
cross/exp1/baseline/my-sw /lm/btec.binlm.1 order=5

# dense weights for feature functions
[weight]
# The default weights are NOT optimized for translation quality. You MUST tune the
weights.
# Documentation for tuning is here:
http://www.statmt.org/moses/?n=FactoredTraining.Tuning
```

```
UnknownWordPenalty0= 1
WordPenalty0= -1
PhrasePenalty0= 0.2
TranslationModel0= 0.2 0.2 0.2 0.2
LexicalReordering0= 0.3 0.3 0.3 0.3 0.3 0.3
Distortion0= 0.3
LM0= 0.5
```

All these processes can be run for nine steps with a single command. An appropriate directory is created as follows, and then run the training command, catching logs:

```
nohup nice ~/mosesdecoder/scripts/training/train-model.perl -root-dir train \
-corpus ~/experiment/msl/msw.true.clean \
-f sw -e my -alignment grow-diag-final-and -reordering msd-bidirectional-fe \
-lm 0:3:$HOME/experiment/msl/msl.blm.en:8 \
-external-bin-dir ~/mosesdecoder/tools >& training.out &
```

Once it's finished there should be a moses.ini file in the directory ~/working/train/model. The first is that it's very slow to load, but we can fix that by binarising the phrase table and reordering table, i.e. compiling them into a format that can be load quickly. The second problem is that the weights used by Moses to weight the different models against each other are not optimized - if you look at the moses.ini file you'll see that they're set to default values like 0.2, 0.3 etc. To find better weights we need to tune the translation system, which leads us on to the next step.

5.4.5. Building a Language Model

The language model (LM) is used to ensure fluent output, so it is built with the target language (i.e SignWriting in this case). The KenLM documentation gives a full explanation of the command line options, but the following will build an appropriate five-gram language model.

```
~/mosesdecoder/bin/lmplz -o 3 \
<~/experiment/msl/msw.true.my > \
~/experiment/msl/msw.arpa.en
```

Then you should binaries (for faster loading) the *.arpa.en file using KenLM:

```
~/mosesdecoder/bin/build_binary\
~/experiment/msl/msw.arpa.en \
~/experiment/msl/msw.blm.en
```

5.5. Tuning

The key to good translation performance is having a good phrase translation table. But some tuning can be done with the decoder. The most important is the tuning of the model parameters.

The probability cost that is assigned to a translation is a product of probability costs of four models:

- phrase translation table
- language model
- reordering model
- word penalty.

Each of these models contributes information over one aspect of the characteristics of a good translation:

- The phrase translation table ensures that the Myanmar words and the SignWriting words are good translations of each other.
- The language model ensures that the output is fluent.
- The distortion model allows for reordering of the input sentence, but at a cost: The more reordering, the more expensive is the translation.
- The word penalty ensures that the translations do not get too long or too short.

This is the slowest part of the process, so you might want to line up something to read whilst it's progressing. Tuning requires a small amount of parallel data, separate from the training data.

```
nohup nice ~/mosesdecoder/scripts/training/mert-moses.pl \
~/experiment/msl/msw.true.sw ~/experiment/msl/msw.true.my \
~/mosesdecoder/bin/moses train/model/moses.ini --mertdir ~/mosesdecoder/bin/ \
&> mert.out &
```

The end result of tuning is an ini file with trained weights, which should be in ~/working/mertwork/.

5.6. Testing

After finishing the previous stages, we can now run the Moses with the following command:

```
~/mosesdecoder/bin/moses -f ~/experiment/msl/model/moses.ini
```

and type Myanmar sentence to see the results.

```
dell@dell-pc:~/experiment/34.my-sw-cross/exp1/baseline/my-sw/model$
~/mosesdecoder/bin/moses -f moses.ini.1
Defined parameters (per moses.ini or switch):
        config: moses.ini.1
        distortion-limit: 6
        feature: UnknownWordPenalty WordPenalty PhrasePenalty
PhraseDictionaryMemory name=TranslationModel0 num-features=4
path=/home/hninwai/experiment/34.my-sw-cross/exp1/baseline/my-sw/model/phrase-
table.1.gz input-factor=0 output-factor=0 LexicalReordering
name=LexicalReordering0 num-features=6 type=wbe-msd-bidirectional-fe-allff
input-factor=0 output-factor=0 path=/home/hninwai/experiment/34.my-sw-
cross/exp1/baseline/my-sw/model/reordering-table.1.wbe-msd-bidirectional-fe.gz
Distortion KENLM name=LM0 factor=0 path=/home/hninwai/experiment/34.my-sw-
cross/exp1/baseline/my-sw/lm/btec.binlm.1 order=5
        input-factors: 0
        mapping: 0 T 0
        weight: UnknownWordPenalty0= 1 WordPenalty0= -1 PhrasePenalty0= 0.2
TranslationModel0= 0.2 0.2 0.2 0.2 LexicalReordering0= 0.3 0.3 0.3 0.3 0.3 0.3
Distortion0= 0.3 LM0= 0.5
line=UnknownWordPenalty
FeatureFunction: UnknownWordPenalty0 start: 0 end: 0
line=WordPenalty
FeatureFunction: WordPenalty0 start: 1 end: 1
line=PhrasePenalty
FeatureFunction: PhrasePenalty0 start: 2 end: 2
line=PhraseDictionaryMemory name=TranslationModel0 num-features=4
path=/home/hninwai/experiment/34.my-sw-cross/exp1/baseline/my-sw/model/phrase-
```

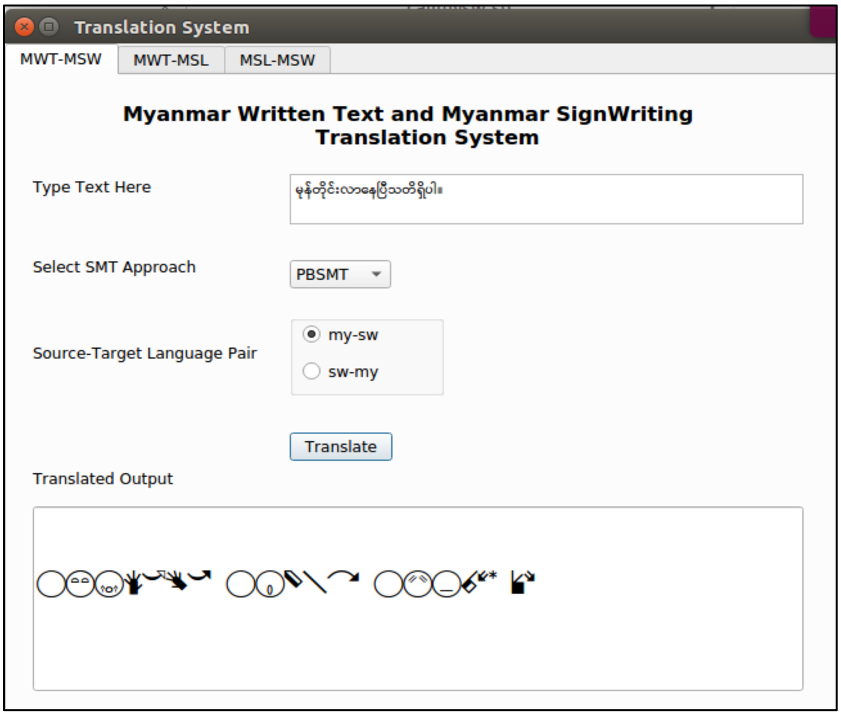
```
table.1.gz input-factor=0 output-factor=0
FeatureFunction: TranslationModel0 start: 3 end: 6
line=LexicalReordering name=LexicalReordering0 num-features=6 type=wbe-msd-
bidirectional-fe-allff input-factor=0 output-factor=0
path=/home/hninwai/experiment/34.my-sw-cross/exp1/baseline/my-
sw/model/reordering-table.1.wbe-msd-bidirectional-fe.gz
Initializing Lexical Reordering Feature...
FeatureFunction: LexicalReordering0 start: 7 end: 12
line=Distortion
FeatureFunction: Distortion0 start: 13 end: 13
line=KENLM name=LM0 factor=0 path=/home/hninwai/experiment/34.my-sw-
cross/exp1/baseline/my-sw/lm/btec.binlm.1 order=5
FeatureFunction: LM0 start: 14 end: 14
Start loading text phrase table. Moses format: [0.116] seconds
Reading /home/hninwai/experiment/34.my-sw-cross/exp1/baseline/my-
sw/model/phrase-table.1.gz
----5---10---15---20---25---30---35---40---45---50---55---60---65---70---75---80---85--
-90---95--100
****************************************************************************************************
Created input-output object: [0.186] seconds
```

The decoder is controlled by the configuration file moses.ini.

5.7. System Implementation

The implementation of this research is set up on Ubuntu desktop computers running Ubuntu 16.04 LTS Linux OS. Moses SMT toolkit is installed for training PBSMT, HPBSMT and OSM statistical machine translation systems. The graphical user interface is implemented with python programming language for translation between MWT-MSW, MWT-MSL and MSL-MWT.

The implementation of the translations can be seen in Fig 5.10 and Fig 5.11 for MWT-MSW translation. The MWT-MSL translation are shown in Fig 5.12 and Fig 5.13 and MSL-MSW translation in Fig 5.14 and Fig 5.15.



Translation System

MWT-MSW | MWT-MSL | MSL-MSW

Myanmar Written Text and Myanmar SignWriting Translation System

Type Text Here: မုန်တိုင်းလာနေပြီသတိရှိပါ။

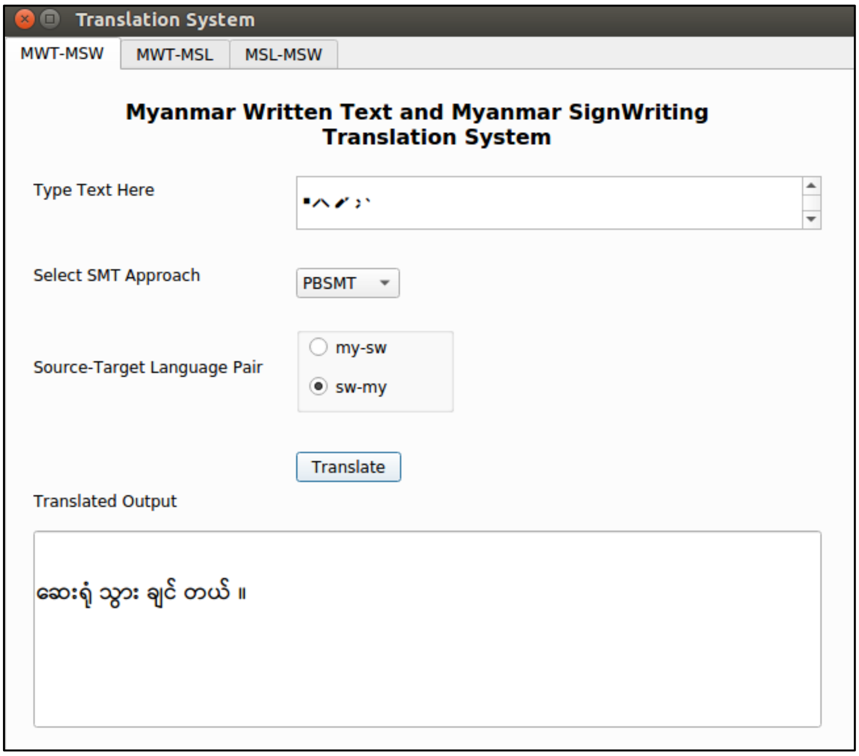
Select SMT Approach: PBSMT

Source-Target Language Pair: ● my-sw ○ sw-my

Translate

Translated Output

Figure 5.10. Machine translation from MWT to MSW using PBSMT approach



Translation System

MWT-MSW | MWT-MSL | MSL-MSW

Myanmar Written Text and Myanmar SignWriting Translation System

Type Text Here

Select SMT Approach: PBSMT

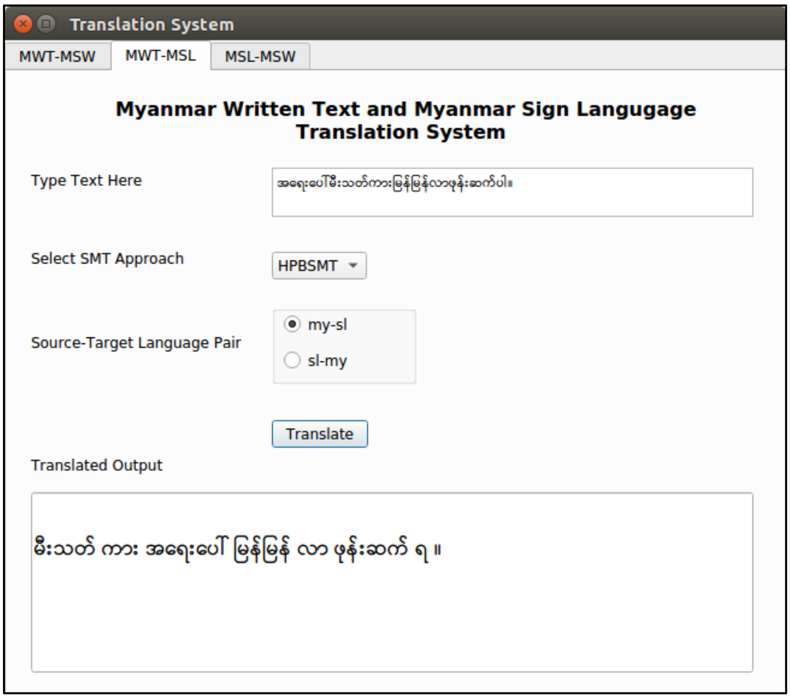
Source-Target Language Pair: ○ my-sw ● sw-my

Translate

Translated Output

ဆေးရုံ သွား ချင် တယ် ။

Figure 5.11. Machine Translation from MSW to MWT using PBSMT approach



Myanmar Written Text and Myanmar Sign Langugage Translation System

Type Text Here: အရေးပေါ်မီးသတ်ကားမြန်မြန်လာဖုန်းဆက်ပါ။

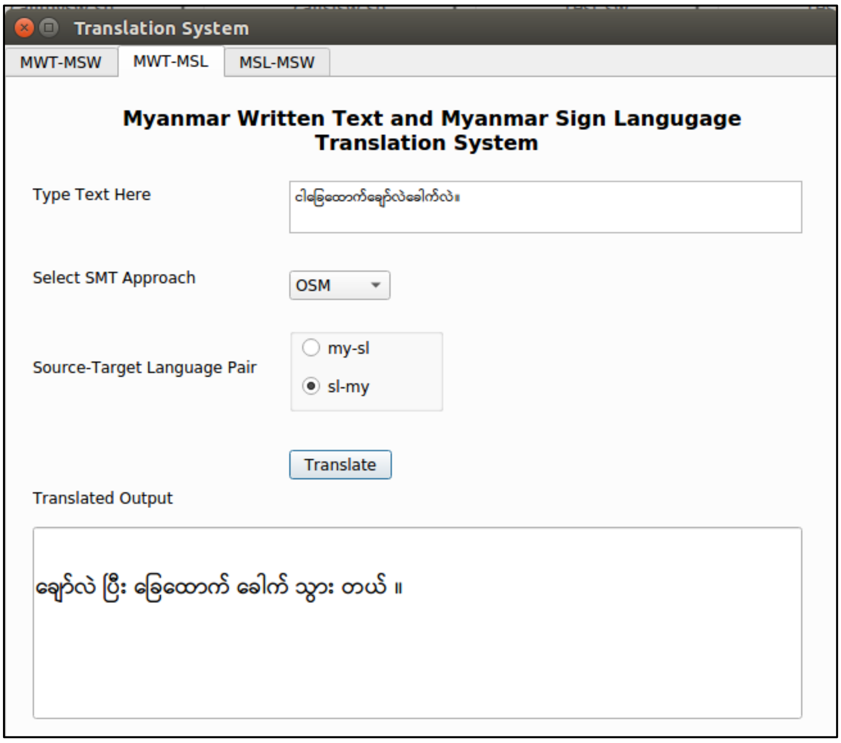
Select SMT Approach: HPBSMT

Source-Target Language Pair: ● my-sl ○ sl-my

Translated Output:

မီးသတ် ကား အရေးပေါ် မြန်မြန် လာ ဖုန်းဆက် ရ ။

Figure 5.12. Machine translation from MWT to MSL using HPBSMT approach



Myanmar Written Text and Myanmar Sign Langugage Translation System

Type Text Here: ငါခြေထောက်ချော်လဲခေါက်လဲ။

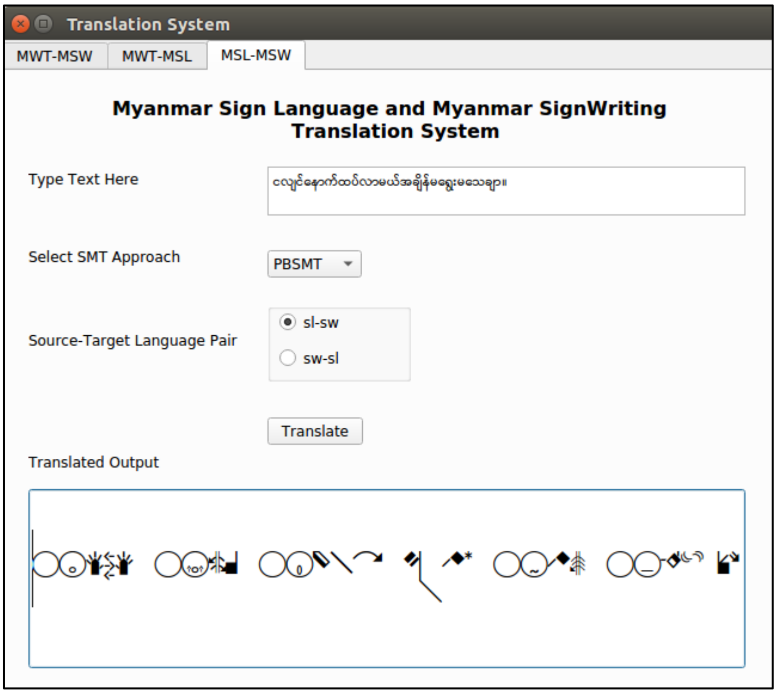
Select SMT Approach: OSM

Source-Target Language Pair: ○ my-sl ● sl-my

Translated Output:

ချော်လဲ ပြီး ခြေထောက် ခေါက် သွား တယ် ။

Figure 5.13. Machine translation from MSL to MWT using OSM approach



Myanmar Sign Language and Myanmar SignWriting Translation System

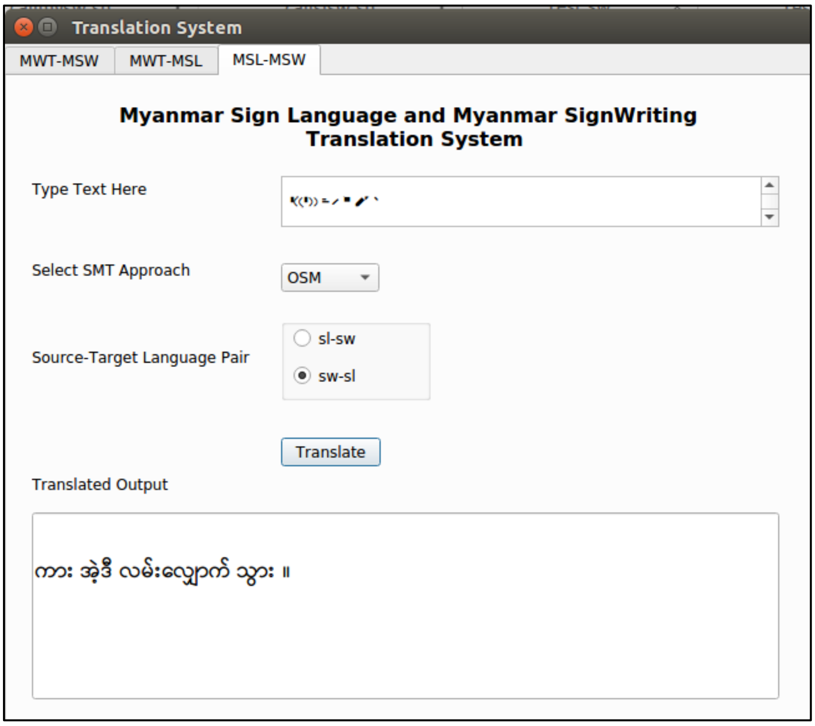
Type Text Here: ငလျင်နောက်ထပ်လာမယ်အချိန်မရွေးမသေချာ။

Select SMT Approach: PBSMT

Source-Target Language Pair: ● sl-sw ○ sw-sl

Translated Output

Figure 5.14. Machine translation from MSL to MSW using PBSMT approach



Myanmar Sign Language and Myanmar SignWriting Translation System

Type Text Here

Select SMT Approach: OSM

Source-Target Language Pair: ○ sl-sw ● sw-sl

Translated Output: ကား အဲ့ဒီ လမ်းလျှောက် သွား ။

Figure 5.15. Machine translation from MSW to MSL using OSM approach

5.8. Summary

This chapter presents the step-by-step procedures of machine translation. The statistical approaches are used to build models that will be used for training, tuning and testing of experiments. And then, the structure of three SMT approaches and system implementation are expressed with examples. The basic requirements and experimental setup are described in this chapter.

CHAPTER 6
EVALUATION AND ANALYSIS

In this chapter, the performance of proposed three SMT systems has been analyzed. The overall performance and error are evaluated for PBSMT, HPBSMT and OSM. The experimental results and error analysis are shown for MWT-MSW translation, MWT-MSL translation and MSL-MSW translation in this chapter.

6.1. Corpus Statistics

This research used 888 Myanmar Written Text (MWT), Myanmar Sign Language (MSL) and Myanmar SignWriting (MSW) parallel sentences, which is a collection of emergency domain, as shown in Section 5.2. In this experiment, three segmentation schemes are applied: syllable segmentation, word segmentation and sign unit-based segmentation. The detail explanation is in Section 5.2.2. In addition, 600 sentences are used for training data, 138 sentences for development data and 150 sentences for testing data.

6.2. Evaluation

Two automatic criteria are used for the evaluation of the machine translation output. One was the de facto standard automatic evaluation metric Bilingual Evaluation Understudy (BLEU) [02Pap] and the other was the Rank-Based Intuitive Bilingual Evaluation Measure (RIBES) [10Iso].

The BLEU score measures the adequacy of the translations and RIBES is suitable for distance language pairs such as Myanmar and English. The higher BLEU and RIBES scores are better. The current parallel corpus size is limited and thus 10-fold cross validation was done for all experiments. The cross validation structure is shown in Fig 6.1. Cross validation means the data are shuffle and divided into 10 times. And then, the first file is used as test data, the second file is used as development data and the remaining as training data in the first experiment. In the same way, the second file is used as test data, the third file is used as development data and the remaining as the training data. Ten experiments are done for all ten files.

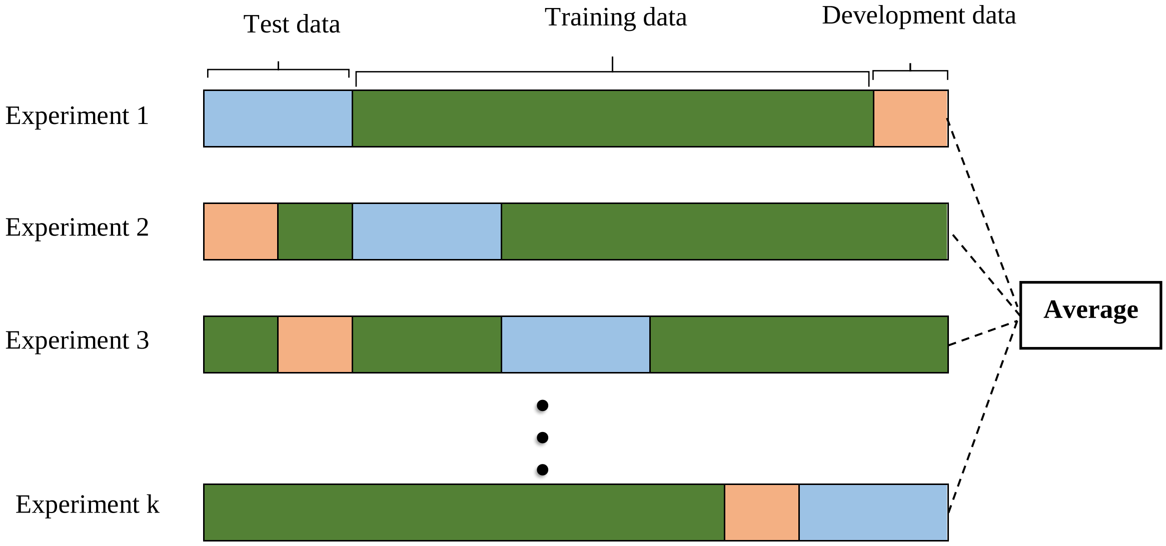


Figure 6.1 Cross Validation Structure

6.2.1. Example Calculation of BLEU

In this example calculation, the parallel data are used: sign language as hypothesis and Myanmar text as reference.

Hypothesis: ကျွန်တော့် အစ်ကို ပျောက် သွား တယ် ကို ရှာ ပေး ပါ ။

References: ကျွန်တော့် အစ်ကို ကို ရှာ ပေး ပါ ။

In this case, c is the length of hypothesis and r is the length of reference sentence. The length of hypothesis sentence is longer than the length of reference sentence, and thus it satisfies c > r, so the value of brevity penalty is set to one. It used 4-gram and uniform weight $w_n = \frac{1}{N}$.

Table 6.1. Example of BLEU Calculation

N-gram	**W_n**	**P_n**	**Log P_n**	**W_n * Log P_n**
1	1/4	7/10	-0.1549	-0.0387
2	1/4	5/9	-0.2553	-0.0638
3	1/4	3/8	-0.426	-0.1065
4	1/4	2/7	-0.5441	-0.136
Total				-0.345
BLEU = 1* exp(-0.345) = **0.7082** =**70.82%**				

6.2.2. Example Calculation of RIBES

Suppose it has the reference and hypothesis sentence for MSW. MSW sentences are transcribed with Myanmar text in this example.

R0: ကျွန်တော့် အစ်မ လှေကား ပေါ်က လိမ့်ကျ ဆေးရုံတင် ။

H0: ငါ့ လှေကား ပေါ်က ချော်ကျ အစ်မ သွား ။

By removing non-aligned words by one-to-one correspondence,

R0: ကျွန်တော့်$_1$ အစ်မ$_2$ လှေကား$_3$ ပေါ်က$_4$ လိမ့်ကျ$_5$ ။$_6$

H0: ငါ့$_1$ လှေကား$_3$ ပေါ်က$_4$ ချော်ကျ$_5$ အစ်မ$_2$ ။$_6$

Word order of R1: [1,2,3,4,5,6]

Word order of H1: [1,3,4,5,2,6]

Number of Increasing pairs = 12

Number of all pairs = 15

Kendall's τ $=2*\frac{12}{15} - 1 = 0.6$

Normalized Kendall's τ (NKT) = $\frac{(0.6+1)}{2}$ = **0.8**

6.3. Experiments

The two automatic criteria are used for the evaluation of the machine translation output. Two segmentation schemes are applied for all experiments, as shown in Section 5.2. The BLEU and RIBES score results for machine translation experiments with PBSMT, HPBSMT and OSM. The values in brackets are RIBES scores. Bold numbers indicate the highest scores among the three SMT approaches.

6.3.1. MWT and MSW Experimental Results

The first experiments are done for Myanmar written text and Myanmar SignWriting (MWT-MSW) translation. Two segmentation schemes are applied for MWT, previously described in Section 5.2.2. The corpus size used in this experiment is 888 parallel sentences for emergency domain. The experimental results between MWT (word) and MSW (word) are shown in Table 6.2.

Looking at the results in Table 6.2 and Table 6.3, MWT (word)-MSW segmentation scheme was by far the most effective for both MWT-MSW and MSW-MWT translation.

Table 6.2. BLEU and RIBES Scores of Myanmar Written Text (word) and Myanmar SignWriting pairs for PBSMT, HPBSMT and OSM

src-trg	**MWT(word)-MSW**		
	PBSMT	**HPBSMT**	**OSM**
my-sw	**12.476** (0.6526)	12.328 (0.6525)	12.427 **(0.6498)**
sw-my	22.268 (0.7184)	22.209 (0.7281)	**23.063** **(0.7295)**

Table 6.3. BLEU and RIBES Scores of Myanmar Written Text (syllable) and Myanmar SignWriting pairs for PBSMT, HPBSMT and OSM

src-trg	**MWT(syllable)-MSW**		
	PBSMT	**HPBSMT**	**OSM**
my-sw	9.108 (0.6358)	**9.432** **(0.6388)**	9.273 (0.6289)
sw-my	20.345 (0.7231)	20.081 (0.7234)	**21.628** **(0.7288)**

In Table 6.2, MWT-MSW translation achieved the highest BLEU and RIBES scores (12.476 and 0. 6498) in PBSMT approach and MSW-MWT translation achieved the highest BLEU and RIBES scores (23.063 and 0.7295) in OSM approach. From the overall results, it can be clearly seen that PBSMT approach is better for both MWT to MSW and MSW to MWT translation. OSM and HPBSMT results are comparable for both word and syllable segmentations. If only focus on syllable segmentation experiments, all three SMT approaches, PBSMT, HPBSMT and OSM results are comparable.

6.3.2. MWT and MSL experimental results

The previous experiments are done with direct translation of MWT-MSW pairs. In building MWT-MSW parallel corpus, MSL concerns as an intermediate between MWT and MSW. Therefore, the experimental results are tested for MWT and MSL parallel data. Now the experiments are started with two segmentations for MWT and MSL: syllable and word segmentation for MWT, and syllable and sign unit-based segmentation for MSL. The experimental results between MWT(word)-MSL(word) are shown in Table 6.4.

Table 6.4. BLEU and RIBES Scores of Myanmar written text (word) and Myanmar sign language(word) pairs for PBSMT, HPBSMT and OSM

src-trg	**MWT(word)-MSL(word)**		
	PBSMT	**HPBSMT**	**OSM**
my-sl	27.359 (0.7436)	**28.825** **(0.7591)**	27.376 (0.7398)
sl-my	29.1 (0.7772)	29.1 (0.7836)	**30.604** **(0.7675)**

Table 6.5. BLEU and RIBES Scores of Myanmar Sign Language (syllable) and Myanmar sign language(syllable) pairs for PBSMT, HPBSMT and OSM

src-trg	**MWT(syllable)-MSL(syllable)**		
	PBSMT	**HPBSMT**	**OSM**
my-sl	31.877 (0.8348)	**32.19** **(0.8352)**	32.125 (0.8342)
sl-my	28.879 (0.8384)	30.414 (**0.8421**)	**30.767** (0.8391)

In table. 6.4, MWT-MSL translation achieved the highest BLEU and RIBES scores (28.825 and 0.7591) using HPBSMT approach and MSL-MWT translation gave the highest BLEU and RIBES scores (30.604 and 0.7675) in PBSMT. For syllable segmentation experiments in Table 6.5, it achieved the highest scores in OSM for both MWT-MSL and MSL-MWT translation pairs. From the overall results, OSM approach is better for both MSL- MSW and MSW- MSL translations.

6.3.3. MSL and MSW experimental results

In this experiment, Myanmar sign language and Myanmar SignWriting translation are done for syllable segmentation and sign unit-based segmentation.

Table 6.6. BLEU and RIBES Scores of Myanmar Sign Language (word) and Myanmar SignWriting(word) pairs for PBSMT, HPBSMT and OSM

src-trg	**MSL(word)-MSW(word)**		
	PBSMT	**HPBSMT**	**OSM**
sl-sw	**41.122** **(0.8577)**	40.82 (0.8553)	40.823 (0.8567)
sw-sl	**50.184** (0.8759)	50.073 **(0.8764)**	49.497 (0.8741)

Table 6.7. BLEU and RIBES Scores of Myanmar Sign Language (syllable) and Myanmar SignWriting (word) pairs for PBSMT, HPBSMT and OSM

src-trg	MSL(syllable)-MSW(syllable)		
	PBSMT	**HPBSMT**	**OSM**
sl-sw	32.094 (0.8298)	**33.069** **(0.8314)**	31.554 (0.8274)
sw-sl	47.674 (0.8674)	47.703 (0.8692)	**48.125** **(0.8695)**

In word segmentation in Table 6.6, PBSMT approach get the higher score in both MSL-MSW and MSW-MSL translations. In syllable segmentation in Table 6.7, HPBSMT and OSM approaches are comparable. From the overall experimental results, word segmented MSL get the better performance than the syllable segmented MSL.

6.3. Error Analysis

The detail explanation related with WER is explained in section 4.12. The following example shows the WER calculation on the translated outputs of three SMT approaches for MSW-MWT language pair with the word segmentation approach. In this example, S =3, D=2, C =5, N= 10 for PBSMT and HPBSMT, its WER is equal to 50%.

Ref: ငါ အရေးပေါ် လူနာတင် ကား လိုချင် လို့ အမြန် ခေါ် ပါ ။

Hyp: ငါ အရေးပေါ် လူနာတင် ကား အမြန် ခေါ်ပေး လိုချင် ။

WER errors:

Reference	**Hypothesis**	**Error Type**
လိုချင်	အမြန်	Substitution
လို့		Deletion
အမြန်	ခေါ်ပေး	Substitution
ခေါ်	လိုချင်	Substitution
ပါ		Deletion

The following example shows the WER calculation on the MWT-MSW where the word segmentation. In this case, C=7, S=4, D=5 for PBSMT and OSM, its WER is equal to 56.25% and C=7, S=4, D=5, I=1 for HPBSMT and its WER is equal to 62.5%.

Ref:	သစ်ပင် ဓာတ်ကြိုး ဖုန်းကြိုး တွေ နား မှာ ရပ် မ နေ နဲ့ ဝေးဝေး ရှောင် ပါ ။
PBSMT Hyp:	လျှပ်စစ် ကြိုး များ သစ်ပင် ဖုန်း ရပ် နဲ့ ဝေးဝေး ရှောင် ပါ ။
HPBSMT Hyp:	သစ်ပင် ဖုန်း လျှပ်စစ် ကြိုး များ ရပ် နဲ့ ဝေးဝေး နေ ပါ ။
OSM Hyp:	လျှပ်စစ် ကြိုး များ ရပ် သစ်ပင် ဖုန်း နဲ့ ဝေးဝေး ရှောင် ပါ ။

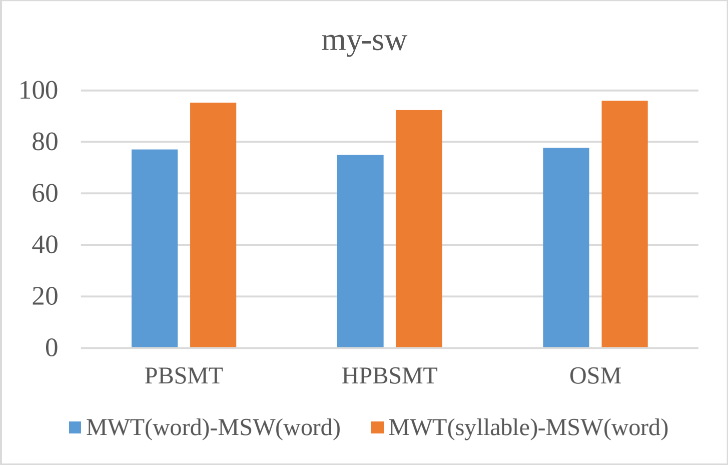


Figure 6.2. WER of Machine Translation between Myanmar Written Text and Myanmar SignWriting

The results show that MWT(word)-MSW segmentation pairs gave the lowest WER values and the difference is higher for the MWT-MSW translation. WER results for MSL-MSW and MSW-MSL translations are shown in Fig 6.2 and Fig 6.3. According to Fig 6.3, WER of word segmented and syllable segmented MWT and MSW pairs achieved the similar scores in WER values.

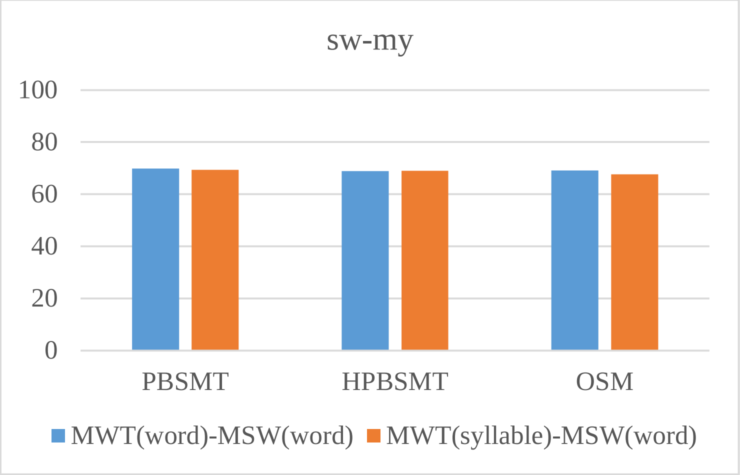


Figure 6.3. WER of Machine Translation between Myanmar SignWriting and Myanmar Written Text

According to the Fig 6.4 and Fig 6.5, WER result of syllable segmented MWT and MSL achieved the lowest scores in both MWT-MSL and MSL-MWT translations. But, the difference is that the WER results in Fig 6.6 and Fig 6.7 get the lowest score in word segmented MSL and MSW, in both directions.

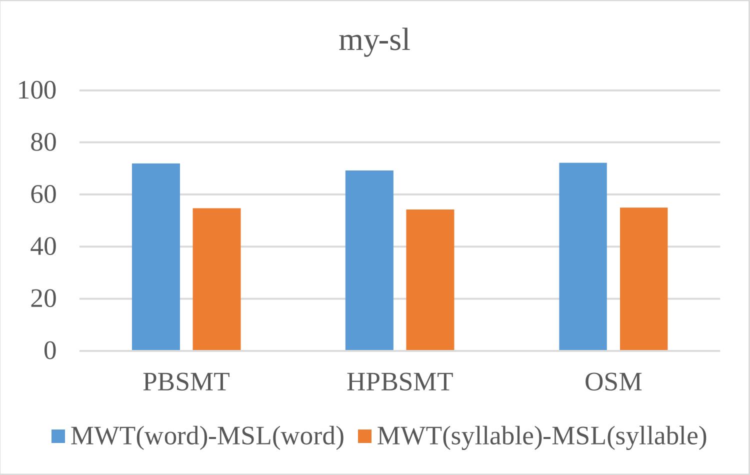


Figure 6.4. WER of Machine Translation between Myanmar Written Text and Myanmar Sign Language

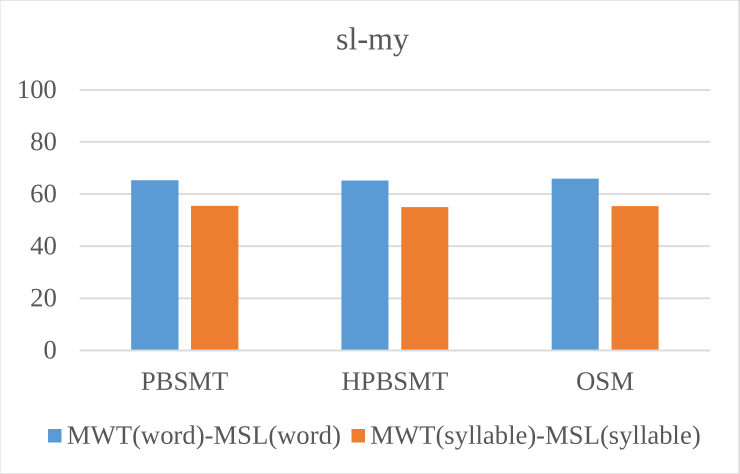


Figure 6.5. WER of Machine Translation between Myanmar Sign Language and Myanmar Written Text

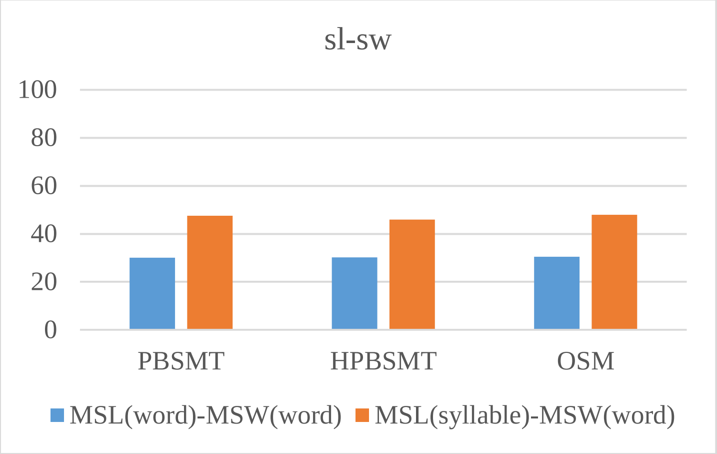


Figure 6.6. WER of Machine Translation between Myanmar Sign Language and Myanmar SignWriting

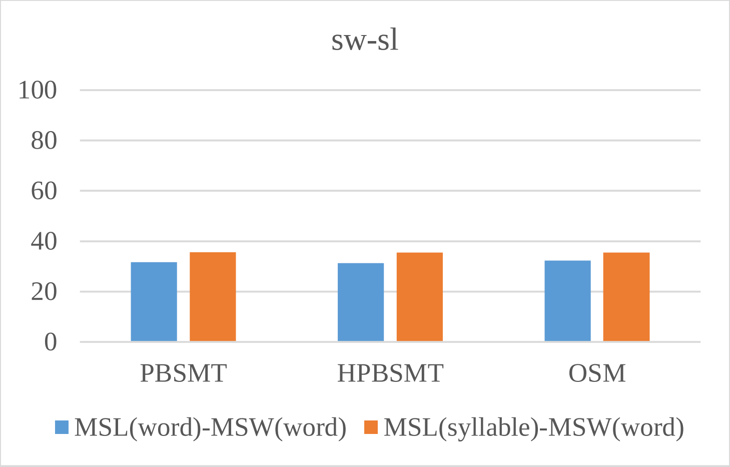
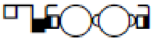
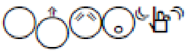
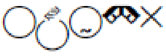
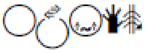
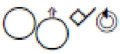
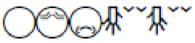
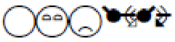
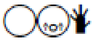
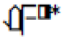
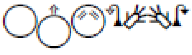
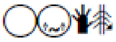
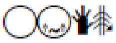
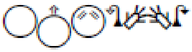
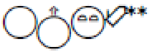
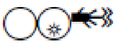
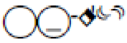


Figure 6.7. WER of Machine Translation between Myanmar SignWriting and Myanmar Sign Language

The top 10 confusion pairs of PBSMT translation model is shown in Table 6.8 and Table 6.9. In this Table, the 1st column is the reference, the 2nd column is the hypothesis (i.e. output of the OSM and PBSMT translation model), the 3rd column is the description of reference and hypothesis in Myanmar written text and the 4th column is the frequency of confusion pairs. From detail analysis on confusion pairs of three SMT approaches, most of the confusion pairs are caused by the three main reasons and they are (1) the nature of the sign language (2) some errors in the reference or human mistakes (3) limited size of the training data. Confusion pair number eight, nine and 10 are caused by translation errors. In table 6.8, confusion pair number one, four and five are caused by the same sign language usage for the several meanings.

Table 6.8. Top 10 Confusion Pairs of OSM Model in MWT-MSW Translation

No	References	Hypothesis	Ref-Hyp Description in Myanmar Language	Freq
1 2 3 4 5 6 7 8 9 10	       	       	ဆရာဂန် -> ဆရာဂန် ဘယ်လောက်လဲ -> ဘာလဲ မသုံးနဲ့ -> မလုပ်နဲ့ မလုပ်နဲ့ -> မလုပ်နဲ့ ဘယ်သူ -> ဘာလဲ ကယ်ဆယ် -> ကျေးဇူးပြု၍ ညစ်ပတ် -> ရေ ရှိ -> မသေချာ	3 3 3 3 3 2 2 2 2 1

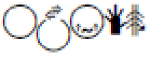
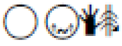
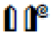
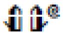
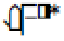
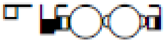
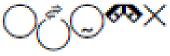
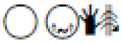
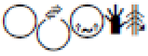
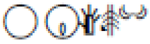
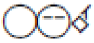
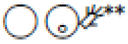
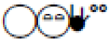
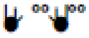
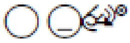
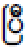
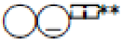
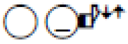
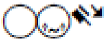
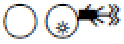
In Myanmar sign language, the word မဟုတ် ("No" in English), the phrase မလုပ်နဲ့ ("Don't do it!" in English), the word မသုံးနဲ့ ("do not" in English) and the phrase မဖြစ် ("Don't use it!" in English) are the same. And thus, the translation model couldn't learn well and it can be assumed this is also relating to the limited size of training data. The confusion pair number two is caused by the error of the reference data. The confusion pair number three and 10 are caused by the sign language dialects (i.e. the difference between Yangon and Mandalay cities sign languages).

One more good example of the confusion pair caused by the sign language nature is the number seven confusion pair, as shown in Table. 6.9. Although one hand sign is using in the reference of the confusion pair number seven, the hypothesis is using two hands.

From the investigation on the overall experimental results, the direct translation of MWT-MSW pair gets the lower performance than MWT-MSL and MSL-MSW translations. In addition, by comparing MWT-MSL and MSL-MSW experimental results, MWT-MSL pair achieved the lower result than MSL-MSW pair.

It can clearly be seen that it is caused by the difference of grammar structure between MWT-MSL and MSL-MSW.

Table 6.9. Top 10 Confusion Pairs of PBSMT Model in MSL-MSW Translation

No	References	Hypothesis	Ref-Hyp Description in Myanmar Language	Freq
1			မလုပ်နဲ့ -> မလုပ်နဲ့	7
2			ရထား -> ရထား	4
3			ဆရာဝန် -> ဆရာဝန်	4
4			မသုံးနဲ့ -> မလုပ်နဲ့	3
5			မလုပ်နဲ့ -> မလုပ်နဲ့	3
6			နည်းနည်း -> ခဏ	3
7			ခံလိုက်ရ -> ပြီ	3
8			အလုပ် -> အလုပ်	3
9			ရေ -> ရေ	3
10				3

6.4. Summary

In this thesis, the statistical machine translation between Myanmar written text and Myanmar SignWriting is implemented and the emergency situations are chosen as a domain area. In this chapter, the three statistical approaches are analyzed using phrase-based statistical machine translation, hierarchical phrase-based statistical machine translation and operation sequence model. In addition, three segmentation schemes are applied: syllable segmentation, word segmentation and sign unit-based segmentation. The experimental results are tabulated comparing three SMT approaches.

CHAPTER 7
CONCLUSION

The statistical approaches are the very widely used research area in machine translation. There are a few studies on automatic machine translation for Myanmar language as well as Myanmar sign language. For this reason, this research focused on the statistical machine translation between Myanmar written text and Myanmar SignWriting. As a result, the collected Myanmar written text, Myanmar sign language and Myanmar SignWriting parallel corpus are obtained. And then, this research leads to easy communication among Deaf and hearing people and gets good knowledge from using SignWriting text symbols, especially in emergency situations.

7.1. Conclusion

This research introduced the first study of the statistical machine translation between Myanmar written text and Myanmar SignWriting. The three SMT approaches (PBSMT, HPBSMT and OSM) are implemented with the current developing MWT-MSW parallel corpus, mainly focused on emergency domain. Three-word segmentation schemes for Myanmar written text and Myanmar Sign Language are investigated in this experiment. In advance, there are many other types of segmentation for Myanmar language, more and more experiments can apply with this data. According to the current experimental result, it clearly shows that OSM approach achieved the highest translation performance for MWT to MSW translation. However, the HPBSMT approach achieved the highest translation performance for MSW to MWT translation. From investigation on the effectiveness of word segmentations for MWT-MSW machine translations, the results proved that word segmentation is better than syllable segmentation for MWT.

7.2. System Limitation

This research is only focused for emergency situation. Therefore, it can only test with Myanmar sentences related with emergency. In addition, there have the font limitation in this research. Since the parallel data are prepared with Myanmar3 Unicode font, the input sentence must be written with the same font. If not, the system does not work well. And then, sign language needs very exact meaning of Myanmar sentences.

7.3. Future Work

The current version of MWT-MSW parallel data is very limited; this research plans to expand this parallel data for more machine translation experiments. In the experiments, segmentation approaches are applied only for Myanmar written text and SignWriting segmentation is just word level. In future work, this research can conduct experiments on SMT with SignWriting character level (i.e. combination of basic symbol, filling symbol and spatial rotation symbol as one SignWriting character) segmentation approach. If more data we can have, it can be planned to investigate neural machine translation, pivot translation compared with SMT approaches. Moreover, the more SignWriting data collected the easier to expand literature and knowledge for Deaf. This will lead to good future for Deaf.

REFERENCES

[17Phi] Philipp Koehn, *Statistical Machine Translation System: User Guide and Code Guide*, (2017)

[17Yek] Ye Kyaw Thu, Syllable Segmentation Tool for Myanmar Language: https://github.com/ye-kyaw-thu/sylbreak

[16Ale] Alex Becker, Fabio Kepler and Sara Candeias.: *A Web Tool for Building Parallel Corpora of Spoken and Sign Languages*, LREC, (2016)

[16Win] Win Pa Pa et al.: *A Study of Statistical Machine Translation Methods for Under Resourced Languages*, 5th Workshop on Spoken Language Technology for Under-Resourced Languages, SLTU, (2016)

[16Yek] Ye Kyaw Thu et al.: A Large-scale Study of Statistical Machine Translation Methods for Myanmar Language, SNLP (2016)

[15Sig] SignMaker 2015 http://www.signbank.org/signmaker/html

[15Win] Win Pa pa et al.: *Word Boundary Identification for Myanmar Text using Conditional Random Fields*, In proceeding of the 9th International Conference on Genetic and Evolutionary Computing, Yangon, Myanmar, (2015)

[14Che] Chenchen Ding et al.: *Empirical Dependency-based Head Finalization for Statistical Chinese-, English-, and French-to-Myanmar (Burmese) Machine Translation*, at the proceeding of the 11th International Workshop on spoken Language Translation, Lak Tahoe, (2014)

[14Nad] Nadir, Fraser, Schmid, Koehn and Schütze.: *The Operation Sequence Model– Combining N-Gram-based and Phrase-based Statistical Machine Translation*, (2014)

[14Cen] *The population and housing census of Myanmar*, (2014)

[13Lew] Lewis et al.: *Deaf Sign Language, Ethnologue: Languages of the World*, SIL International, (2013)

[13Mar] Martin Thoma.: Word Error Rate Calculation, (2013) https://martin-thoma.com/word-error-rate-calculation/

[13Yek] Ye Kyaw Thu et al.: *A Study of Myanmar Word Segmentation Schemes for Statistical Machine Translation*, In Proceedings of the 11th

International Conference on Computer Applications (ICCA), Yangon, Myanmar, (2013)

[12Tin] Tin Htay Hlaing, *Manually constructed context-free grammar for Myanmar syllable structure*, In Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL '12), Association for Computational Linguistics, Stroudsburg, PA, USA

[11Ach] Achraf Othman and Mohamed Jemni.: *Statistical Sign Language Machine Translation: from English written text to American Sign Language Gloss*, International Journal of Computer Science (IJCSI), (2011)

[11Ber] Berichter, Ney, Francisco, *Investigations on Hierarchical Phrase-based Machine Translation*, (2011)

[11Hea] Heafield, Kenneth, *KenLM: Faster and Smaller Language Model Queries*, In the proceedings of the 6th Workshop on Statistical Machine Translation, Association for Computational Linguistics, Edinburgh, Scotland, (2011)

[10Ada] Adam Frost and Valerie Sutton.: *SignWriting Hand Symbols in ISWA2010: Manual 2*

[10Iso] Isozaki, H., Hirao, T., Duh, K., Sudoh, K., Tsukada, H.: Automatic evaluation of translation quality for distant language pairs, In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP ’10, Association for Computational Linguistics, Stroudsburg, PA, USA, (2010) http://dl.acm.org/citation.cfm?id=1870658.1870750

[10Mya] Myanmar Unicode Table, Range:1000–109F, http://www.unicode.org/charts/PDF/U1000.pdf

[10Sut] Valerie Sutton.: *International SignWriting Alphabet (2010)*

[10Val] Valerie Sutton. SignWriting: A complete system for writing and reading signed languages http://www.signwriting.org

[09Koe] Koehn, P.: *Statistical Machine Translation*, Cambridge University Press, (2009)

[08Zin] Zin Maung Maung and Yoshiki Makami.: *A rule-based syllable segmentation of Myanmar Text*, In Proceedings of the IJCNLP-08 workshop of NLP for Less Privileged Language, Hyderabad, India, (2008)

[07Bra] Brants, T et al.: *Large language models in machine translation*, In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLPCoNLL), (2007)

[07Sra] Sra Morrissey and Andy Way.: *Joining Hands: Developing A Sign Language Machine Translation System with and for the Deaf Community*, Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments Assistive Technology for All Ages (CVHI), (2007)

[07Liu] Liu et al.: *Discriminative language model adaptation for Mandarin broadcast speech transcription and translation*, In Proceedings of IEEE Automatic Speech Recognition and Understanding (ASRU), Kyoto, Japan, (2007)

[07Man] *Mandalay School for the Deaf, Myanmar Sign Language Dictionary, (Vol:1 and 2),* (2007)

[06Dan] Daniel Stein, Jan Bungeroth and Hermann Ney; *Morpho-Syntax Based Statistical Methods for Automatic Sign Language Translation*, (2006)

[05Guy] Guylhem Aznar and Patrice Dalle; *SignWriting Unicode support: using an assisted entry process to neutralize signs and symbols variability*, (2005)

[05Ldc] LDC, *Linguistic data annotation specification: assessment of fluency and adequacy of translations*, revision 1.5, (2005)

[04Jan] Jan Bungeroth and Hermann Ney: *Statistical Sign Language Translation*, (2004)

[04Til] Tillmann, C.: *A unigram orientation model for statistical machine translation*, In Proceedings of HLT-NAACL 2004: Short Papers; HLT-NAACL-Short'04. Stroudsburg, PA, USA: Association for Computational Linguistics. ISBN 1-932432-24-8, (2004) http://d1.acm.org/citation.cfm?id-1613984.1614010

[03Koe] Koehn, P., Och, F. J., and Marcu, D, *Statistical phrase-based translation*, In NAACL'03: Proceedings of the 2003 Conference of the

North American Chapter of the Association for Computational Linguistics on Human Language Technology, (2003)

[03Och] Och, F. J, *Minimum error rate training in statistical machine translation*, In ACL'03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, (2003)

[02Ant] Antonio Carlos da Rocha Costa and Gracaliz Pereira Dimuro.: *A SignWriting-Based Approach to Sign Language Processing*, International Gesture Workshop, (2002)

[02Och] Och, F. J. and Ney, H.: *Discriminative training and maximum entropy models for statistical machine translation*, In ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA. Association for Computational Linguistics, (2002)

[02Pap] Papineni, K., Roukos, S., Ward, T., Zhu, W., "Bleu: a Method for Automatic Evaluation of Machine Translation". IBM Research Report rc22176 (w0109022), 2001, Thomas J. Watson Research Center, In ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, (2002)

[00Jur] Jurafsky et al.: Speech *and Language Processing: An Introduction to Natural Language Processing*, Computational Linguistics, and Speech Recognition, Prentice Hall PTR, Upper Saddle River, NJ, USA, (2000)

[00Och] Och, F.J., Ney, H.: *Improved statistical alignment model*, In ACL00. Hong Kong, China, (2000)

[00Mar] Marin, Joe.: *A linguistic comparison Two notation systems for signed languages: Stoke Notation and Sutton SignWriting*, Manuscript on the SignWriting website, (2000)
http://www.signwriting.org/

[99Che] Chen, S. F. and Goodman, J, *An empirical study of smoothing techniques for language modeling*, Computer Speech & Language, (1999)

[96Che] Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling, In Proceedings of the 34th annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, (1996) [93Bro] Brown et al.: *The mathematics of statistical machine translation: parameter estimation*, Computational Linguistics, (1993)

[93Whi] White et al.: *Evaluation of machine translation*, In HLT'93: Proceedings of the workshop on Human Language Technology, pages 206–210, Morristown, NJ, USA. Association for Computational Linguistics, (1993)

[74Val] History of Valerie Sutton
https://en.wikipedia.org/wiki/Valerie_sutton_history

LIST OF PUBLICATIONS

1. Swe Zin Moe, Hlaing Myat Nwe, **Hnin Wai Wai Hlaing**, Ye Kyaw Thu, Hnin Aye Thant, Nandar Win Min, “Myanmar Sign Language (MSL) Corpus for Emergency Domain”, PACLING2017 conference, Yangon, Myanmar. (Demo and Poster)
2. Swe Zin Moe, Ye Kyaw Thu, Hlaing Myat Nwe, **Hnin Wai Wai Hlaing**, Ni Htwe Aung, Hnin Aye Thant, Nandar Win Min, “Corpus Building for Machine Translation between Myanmar Sign Language and Myanmar Written Text”, World Deaf Day 2017, 14th Sept. 2017, Mandalay Community Center, Chan Aye Tharzan Township, Mandalay, Myanmar. (Poster)
3. Swe Zin Moe, **Hnin Wai Wai Hlaing**, Ye Kyaw Thu, Hlaing Myat Nwe, Ni Htwe Aung, Hnin Aye Thant, Nandar Win Min, “မြန်မာ လက်သင်္ကေတပြဘာသာစကားမှ မြန်မာစကားပြောစာကြောင်းသို့ ကွန်ပျူတာသုံး ဘာသာပြန် သုတေသန”, International Day of Persons with Disabilities 2017, 3rd Dec. 2017, Wilson Hotel, No.31(E), Yangon-Mandalay Main Road, Maha Aung Myay Township, Mandalay, Myanmar. (Demo and Poster)
4. Hlaing Myat Nwe, Ye Kyaw Thu, **Hnin Wai Wai Hlaing**, Swe Zin Moe, Ni Htwe Aung, Hnin Aye Thant, Nandar Win Min, "Two Fingerspelling Keyboard layouts for Myanmar SignWriting", International Day of Persons with Disabilities 2017, 3rd Dec. 2017, Wilson Hotel, No.31(E), Yangon-Mandalay Main Road, Maha Aung Myay Township, Mandalay, Myanmar. (Demo and Poster)
5. Hlaing Myat Nwe, Ye Kyaw Thu, **Hnin Wai Wai Hlaing**, Swe Zin Moe, Ni Htwe Aung, Hnin Aye Thant, Nanda Win Min, "Two Fingerspelling Keyboard Layouts for Myanmar SignWriting", In Proceedings of ICCA2018, February 22-23, 2018, Yangon, Myanmar, pp. 290-298. (Paper)
6. Swe Zin Moe, Ye Kyaw Thu, **Hnin Wai Wai Hlaing**, Hlaing Myat Nwe, Ni Htwe Aung, Hnin Aye Thant, Nandar Win Min, "Statistical Machine Translation between Myanmar Sign Language and Myanmar Written Text", In Proceedings of ICCA2018, February 22-23, 2018, Yangon, Myanmar, pp. 217-227. (Paper)
7. **Hnin Wai Wai Hlaing**, Ye Kyaw Thu, Swe Zin Moe, Hlaing Myat Nwe, Ni Htwe Aung, Nandar Win Min, Hnin Aye Thant, “Statistical Machine Translation

between Myanmar Sign Language and Myanmar SignWriting”, at the First IEEE International Symposium on Artificial Intelligence for ASEAN Development, ASEAN-AI2018, Phuket, Thailand, 26th March 2018. (Paper)