EFFICIENT TRANSLATION BETWEEN SINHALA AND TAMIL Using Part of Speech and Morphology

Yashothara Shanmugarasa

(178029B)

Thesis submitted in partial fulfillment of the requirements for the Degree of Master of Science (Research) in Computer Science and Engineering

Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

November 2018

DECLARATION

"I declare that this dissertation has been composed by solely by myself and this dissertation does not combine without acknowledgement of any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

Name: Yashothara.S

The above candidate has carried out research for the Masters Dissertation under my supervision.

Signature of the supervisor: Date:

Name of the supervisor: Prof. Gihan Dias

Signature of the co-supervisor: Date:

Name of the supervisor: Dr. R.T. Uthayasanker

Abstract

Machine translation is the process of translating a document from one language to another with the aid of a computer. Even though many machine translation technologies exist, statistical machine translation (SMT) still provides better performance in terms of quality and time for low resourced languages. In this study, we choose Sinhala to Tamil translation and vice versa since they are official languages of Sri Lanka. Often government official documents are written in one language (the majority in Sinhala) and translated into Tamil. Translation between Tamil and Sinhala is currently a time-consuming manual process carried out by department of official languages. Aiding the translators with an automated translation system will improve the efficiency of this process. However, there are some challenges in statistical machine translation of Tamil and Sinhala.

In the initial part of this research, a study on language divergence was conducted to identify the challenges in machine translation between Tamil and Sinhala. In this thesis, we focus on (i) improving the statistical machine translation from Sinhala to Tamil using a hierarchical phrase-based SMT model, (ii) Parts of Speech (POS) based Factored Statistical Machine Translation system (F-SMT) and (iii) preprocessing techniques based on chunking and segmentation. Based on the analyzed results of language divergence, translation challenges such as (i) reordering, (ii) abbreviations and initials, (iii) word flow of the sentence, (iv) data sparseness, (v) ambiguity in translation, (vi) divergence among Tamil and Sinhala POS tagsets and (vii) mapping one word with one or more words were addressed. We also developed an algorithm for the alignment of different POS tagsets.

Subsequently, we used hierarchical phrase-based model and Factored model with POS integration to address challenges such as word reordering, word flow, context aware word selecting, translating conjunction words, better word choice and translating initials and abbreviations. Further we experimented with some pre-processing techniques based on chunking and segmentation towards addressing challenges such as unknown words, context awareness, better word choice, word flow, ambiguity in translation, translating into proper 'Sandhi' form, translating named entities and replacing one word with multiple words. Point-wise Mutual Information (PMI) based collocation phrases, POS based chunks, Named Entities and sub word segments are used to enhance the preprocessing step. Even though, the standard structure of a sentence is Subject-Object-Verb in both languages, there is a need of reordering in the translation between these languages. As our languages are the low resourced when we try to translate using traditional Statistical machine translation, we are unable to get a good order of sentences because of sub-phrases which have been observed previously in the training corpus can only reorder using distortion reordering model which is independent of their context. To improve reordering, we have tried the hierarchical phrase-based model and

factored model. Hierarchical Phrase-based Model helps to improve translation quality between languages that vary by sentence structure. But it lowers the quality of languages share similar sentence structure and Tamil and Sinhala languages don't have a syntactic parser for better performance.

Parts of speech knowledge is added as the factored model to improve reordering also. The words are factored into lemma and parts of speech. This factored model decreases the data sparseness in decoding and helps to reordering. These linguistic features are considered as separate tokens in the training process. We show that by generalizing translation with parts of speech tags, we could improve performance by 0.74 BLEU on a small Sinhala-Tamil system. Even though we could only achieve small increment in BLEU score, manual evaluation of the translation showed improvements.

Preprocessing is another way of enhancing the quality of the translation. Preprocessing described in the research is related to finding collocation words from PMI, NER based chunking, POS based chunking and segmentation. We observed that each of the preprocessing techniques provided better performance than the baseline system. When comparing the preprocessing methods, PMI based chunking gave good results compared to other preprocessing techniques. A hybrid approach is done by combining preprocessing approaches based on PMI chunking, NER chunking, and POS chunking. BLEU score was increased up to 33.41 by using a hybrid approach. The best performance is reported with hybrid approach for Sinhala to Tamil translation. We could improve performance by 12% BLEU (3.61) using a small Sinhala to Tamil corpus with the help of proposed hybrid approach preprocessing technique. Notably, this increase is significantly higher compared to the increase shown by prior approaches for the same language pair.

Keywords- Statistical Machine Translation, Parts of Speech, POS tagset Mapping, POS tagset Alignment, Semi-Supervised Approach, BIS tagset, UOM tagset, Tamil NLP, Sinhala NLP, Hierarchical Phrase Based model, Parallel corpus

ACKNOWLEDGEMENT

I would never have been able to finish my dissertation without the guidance, support and encouragement of numerous people including my mentors, my friends, colleagues and support from my family. At the end of my thesis I would like to thank all those people who made this thesis possible and an unforgettable experience for me.

First and foremost, I would like to convey my sincere gratitude to my supervisors Dr. R.T. Uthayasanker and Prof. Gihan Dias, for the continuous support given for the success of this research both in unseen and unconcealed ways. This would not have been a success without your tremendous mentorship and advice from the beginning. Their wide knowledge and logical way of thinking have been of great source of inspiration for me. They have always extended their helping hands in solving research problems. I strongly trust that without their guidance, the present work could have not reached this stage.

I wish to thank Dr.Surangika Ranathunga and Prof. Sanath Jayasena for their supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences through-out the work.

This research was supported by Department of Official Languages and University of Moratuwa Senate Research Grant. I sincerely thank colleagues from Department of Official Languages for the support given.

I would like to thank all staff from the Department of Computer Science and Engineering for their kindness expressed in all occasions.

I would like to thank Ms. Nimasha Dilshani and Ms. Fathima Farhath, who as good friends from my graduate is always willing to help and give their best suggestions.

Special thanks to my loving family. Your encouragement always motivated me to do my best. I would also like to thank all my friends who encouraged me all times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AU-KBC – Anna University K B Chandrasekhar

BIS – Bureau Indian Standard

BL – Base Line

BLEU – Bi-Lingual Evaluation Understudy

CIIL – Central Institute of Indian Languages

CRF– Conditional Random Fields

EBMT– Example based Machine Translation

EM – Expectation Maximization

F-SMT – Factored Statistical Machine Translation

HPB – Hierarchical Phrase Based

IIIT– International Institute of Information Technology

MT– Machine Translation

NIST – National Institute of Standards and Technology

NLP – Natural Language Processing

POS– Parts of Speech

RBMT– Rule based Machine Translation

SCFG – Synchronous Context-Free Grammar

SMT – Statistical Machine Translation

SOV– Subject-Object-Verb

SRILM – Stanford Research Institute for Language Modeling

SVM– Support Vector Machine

TER– Translation Edit Rate

TnT– Trigrams n Tagger

UCSC – University of Colombo School of Computing

UOM – University of Moratuwa

WER– Word Error Rate

INTRODUCTION

This thesis focuses on enhancing a statistical machine translation system using the factored model by POS, hierarchical phrase-based and preprocessing techniques. The research addresses two main kinds of preprocessing techniques:

- Preprocessing based on chunking
- Preprocessing based on segmentation

This chapter describes about the overview of machine translation, history, and types of machine translation, importance and applications of machine translation, description of statistical machine translation, the motivation of this thesis, the objective of this thesis and the contributions made from this research.

1.1 Overview of machine translation

Machine Translation is a process of translating documents from one language into another with the aid of a computer. Initial efforts for Machine Translations were made in 1950's. Even though they didn't accomplish what they expected. With the availability of Internet, people got more opportunities to go global. This is where translation plays a major role. As the world becomes more globalized, this problem turned more severe. Human translators are expensive and difficult to find. Machine translation can improve the accuracy of human translators, substitute them completely, or implement the tasks which would have otherwise left incomplete.

Moreover, various communication methods have been developed such as mobile texting, instant messaging, Email, online social media and video conferencing in information society. Machine translation gives the direct and immediate response that would be hard to achieve with human translators.

There are several approaches like Linguistic based and Interlingua based systems to develop machine translation system. A lot of linguistic knowledge such as morphological, syntactic and semantic analysis is required for rule-based approaches during the translation. Transforming text from source language to a common representation is the main aim in Interlingua approach.

But currently, translations using traditional statistical and neural machine translation approaches dominate this field. Statistical machine translation approach combines the different set of knowledge from statistics, data structure, automata

theory, data mining, Natural Language Processing, machine learning and artificial intelligence. In SMT, translation is carried out using a learning algorithm which is applied to a huge amount of manually aligned parallel data. It is kind of a machine learning problem. Parallel corpus is a collection of texts, each of which is translated into one or more other languages than the original [1]. One of the corpora is an exact translation of the other. Parallel corpora for a language pair are important to build a bilingual SMT system. The quality and accuracy of the translation mostly depends on the quality, amount and domain of the parallel data we used to train. How a machine learns the patterns of translation in SMT is described in Figure 1.1.

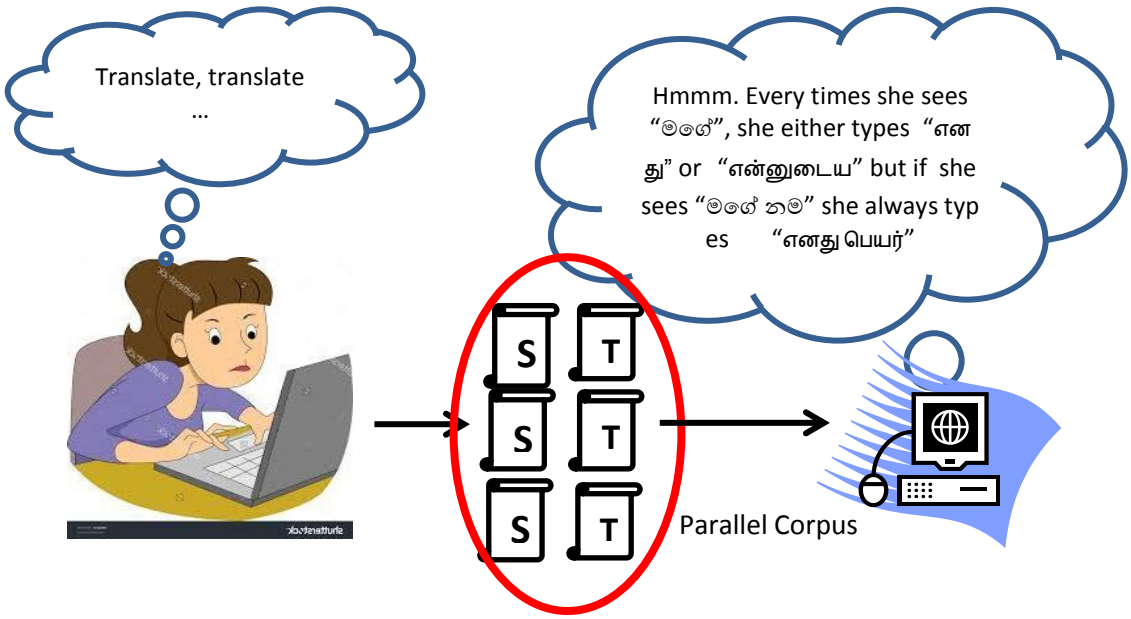


Figure 1.1 Overview of Statistical Machine Translation: Learning Patterns from the parallel corpus

For similar languages in specific domains which have huge parallel corpora, SMT models give good accuracy. The translation patterns are hard to learn if the sentence structures are not similar and with less bilingual data. As a large number of parallel corpora are needed for SMT model, statistical methods are challenging to be used in "low resourced" languages. Both Sinhala and Tamil languages lack in necessary natural language resources and tools, hence classified as low resourced languages. This limits the success achievable in machine translation to and from those languages. To improve the translation accuracy of these low resourced languages, adding linguistic knowledge is required. Linguistic knowledge is added using linguistic tools.

1.1.2 Importance and Application

Though automatic translation systems are not perfect for low resourced languages, now there are number of systems available for translation. Not only word level replacement is considered in the translation. All the elements in the text such as grammar, sentence structure and meanings must be interpreted by the translator. All the issues during the translation should be known to the translation system and those issues should be handled well. The cross-culture understanding is a significant issue that grips the performance of the translation.

So, designing an automatic machine translation system is a great challenge. It is very hard to translate source sentence to target sentence by considering all required information of both languages. Identical translations cannot be generated even with two individual translators. Henceforth, producing high quality automated machine translators is a challenging task.

There are many situations that machine translation will serve for the translation task at hand. Some of the major benefits that can be gained from machine translation are described below.

- Time compatibility: Automatic machine translation is much faster than human translation.
- Minimal cost: Though buying language translating system may look costly early, but it is a much economical solution than spending money on human translation for long time.
- Capability to translate between different languages: One of the great things about machine translation is its ability to translate in many languages, sometimes even hundreds of languages.
- Key terms' memory: A key benefit that comes from machine translation is the fact that translation software has the skill to remember and reuse the common words and phrases that are used within a given domain.
- Web content and web page translation: Web content in web pages can be easily translates.

Different sizes of enterprises are using machine translation applications for different purposes and different level. Multi-domain translation services which mean customizable solutions across different domains are offered by some organizations while other organizations offer translation solutions only for a specific domain. Even though, these solutions are automated, still depend on human translators for editing purposes.

There are many machine applications already available nowadays. Dublin-based KantanMT is a SaaS based machine translation system in the cloud to develop and manage custom translation. According to the company's website, this platform enables the translation services across eight domains such as travel, e-retail, government, etc. SYSTRAN is a machine translation system for five domains/industries. It allows three kinds of models such as Full-text translation, File translation, and web translation services. SDL Government machine translation application serves the US government by focusing on defense and Intel use cases. Canopy Speak is a medical translator app which is based on the pre-translated medical phrases corpus. The corpus is organized by frequently encountered procedures.

Google translate plays a major role among all other machine translation systems. Text, speech and images of words can be easily translated in real time in the Google translate. All these services are packaged into a single platform in the form of a mobile app and cloud service. Facebook has focused on the experiments with machine translation for close to a decade. More sophisticated and intelligent Facebook translation app is evolved based on NMT. For Tamil and Sinhala languages also, there is traditional SMT model called "Si-Ta" which specifically built for the official documents domain.

1.2.4 Problem Definition

Sri Lanka is a multi-ethnic country where Sinhala and Tamil languages are declared as official languages. However, most of the people only know one language due to the longtime war. Yet, often government official documents are written in one language (the majority in Sinhala) and translated into Tamil. However, in order to overcome this language barrier, currently, the support of human translators is used. Yet the requirement of human translators outweighs the supply which leads to incomplete translations and delays in publishing. So, this fails the goal of easing the citizens to

use own language of their preference either in governmental communication or in information seeking. To move forward to a better bilingual communication between the government and the public; the better option is to boost the human translator's efficiency. Already there is a system called “Si-Ta” for the translation among official documents purely based on traditional SMT system. But as I mentioned in the previous section, there are many challenges in the pure traditional method. So, there is a need of more human effort to make the proper translation. ***This research is focused on improving translation by overcoming the issues in the traditional SMT system to improve the system.***

1.3 Motivation of the Thesis

This section discusses the two factors that motivate the research undertaken in this MSc study. First, there is a dearth of research on the improving of traditional Statistical machine translation from Sinhala to Tamil language. Second, there is a need of Tamil to Sinhala translation which is not currently available in the “Si-Ta” system. These two motivations are described in the following sections.

1.3.1 Research motivation

Machine translation is the process of using the computers to translate texts from one natural language to another. Even though, in the 1950’s machine translation was proposed, it is still considered as an open problem and people didn’t find a system which works with 100% accuracy. But, the demand for automatic machine translation grows rapidly due to the globalization.

United Nations put an effort to translate a large number of documents into several languages initially. They have created bilingual corpora for some language pairs like Chinese-English, Arabic-English and distributed through the Linguistic Data Consortium (LDC). Around 20% of web pages are available in their national languages except English. To translate these web pages and resources to the required language machine translation can be used [7].

Sinhala-Tamil translation gains importance since both Sinhala and Tamil are official languages practiced in our country (along with English) but the most of the population can read/write only in one language. Often government official documents are written in one language (mostly in Sinhala) and translated into Tamil. This is a time-consuming and manual process carried out by the department of official

languages on a daily basis. A key bottleneck in improving the efficiency of the translation process is the lack of Tamil - Sinhala translators. Aiding the translators with an automated translation system would improve the efficiency of this process.

Currently, translation of official documents between Sinhala and Tamil languages is done manually. For word processing only, automation is used. Translation of annual reports of public sectors and government departments involves lots of manual effort. Human translation takes more time and cost compared to machine translation. It is clear from this that there is large market value available for machine translation rather than human translation between Sinhala and Tamil languages. As machine translation is faster, comfort and cheaper, most people will to choose machine translation over human translation. It will reduce the effort of a human.

In this study, we choose Tamil and Sinhala languages which gain importance since both of them are acknowledged as official languages of Sri Lanka. Further, since these two languages are considered as low resourced languages, these efforts gain more importance. The Sinhala language belongs to the Indo Aryan language family and the Tamil language belongs to the Dravidian family. As two languages that have been in contact for a long period of time, they share notable resemblances in morphology and syntax. Tamil, a Dravidian language, is spoken by around 72 million people. Tamil is spoken in Sri Lanka, Tamil Nadu, Singapore, Malaysia and Mauritius as well as emigrant communities around the world.

In this thesis, a methodology for improving the statistical machine translation systems from Sinhala to Tamil is proposed. Initially, to identify the issues in the machine translation system, we carried out an analysis of divergence between Sinhala and Tamil languages. Based on the analyzed results of language divergence, a divergence among Tamil and Sinhala POS tagsets could be identified. Accordingly, we have come up with an algorithm for the alignment of different POS tagsets. With the analyzed results of language divergence, we have come up with some techniques to improve the translation. Factored model based on Parts of Speech tagsets and hierarchical model is used to handle reordering between Sinhala and Tamil languages. Preprocessing techniques are used to overcome the challenges such as mapping one

word to more words, out of vocabulary, name entity translation, word flow ambiguity, and context-aware translation.

1.3.2 Si-Ta system motivation

Si-Ta system is developed by the University of Moratuwa for the department of official languages. Si-Ta is a Machine Translation system for Sinhala and Tamil languages which focused on official government documents, with post editing support to correct the translation. As current system is not having Tamil to Sinhala translation, this research focused on developing Tamil to Sinhala translation. Even though most of the documents are originally written in Sinhala language, we need Tamil translation for those documents, North and East provincial documents are originally written in the Tamil language. So there is a need of Tamil to Sinhala translation also.

1.4 Objective of the Thesis

The main objectives of the proposed research are to increase the efficiency of Statistical Machine Translation (SMT) by overcoming the challenges, using POS and some preprocessing techniques from Sinhala to Tamil and increase its applicability for official documents domains. This research will also address the challenges such as word reordering, unknown words, context awareness, better word choice, word flow, ambiguity in translation, translating name entities, translating abbreviation and initials and mapping one word with one or more words between target and source sentences when translating from a morphologically rich language into a morphologically rich language. It also addresses a semi-automatic alignment algorithm for aligning multilingual Parts of Speech tagsets.

1.5 Contributions

The contributions of this research are three-fold. First, the useful additions to the research expedition towards machine translation between Sinhala and Tamil. Second, the research articles I published and presented based on these additions. Third the contributions I made to SiTa system beyond the research outcomes mentioned in the first point.

To address the objectives mentioned in the previous section, a system has been developed and improved with the following capabilities:

- Identified five types of lexical-semantic language divergence between Sinhala and Tamil languages to enhance the quality of translation.
- Based on the analyzed results of language divergence, semi-automatic algorithm to cast the problem of heterogeneity in POS tagsets as an alignment of two labeled trees is proposed.
- To overcome challenges in SMT, studied the impact of Hierarchical phrase-based (HPB) machine translation for low resourced languages and provided recommendations in choosing HPB MT systems based on morphological richness
- Identified a suitable tagset and tagger for Tamil and Sinhala for the integration of POS into SMT.
    - Sinhala Tagset: UOM tagset
    - Tamil Tagset: BIS tagset
    - Sinhala Tagger: UOM POS tagger
    - Tamil Tagger: AUKBC tagger
- Integration of Factored MT with POS into SiTa and improved the translation quality
- Studied a suite of preprocessing methods and identified a best setting which improves the Sinhala to Tamil translation.
- Developed first ever Tamil to Sinhala translation system
- Presented two accepted papers and two more papers have been accepted.

1.5.1. Articles

This research has produced the following refereed publication so far:

- Presented on "Hierarchical Machine Translation Workbench for Indian Languages", 2017 May, IASNLP summer school IIIT, Hyderabad
- Presented submission: Yashothara.S, R.T.Uthayasanker, "A study on the utility of Hierarchical phrase-based model for low resourced languages", 2017, International Conference on Linguistics in Sri Lanka, University of Kelaniya
- Presented submission: Yashothara.S, R. T.Uthayasanker, G.V.Dias "Semi-Automatic Alignment of Multilingual Parts of Speech Tagsets ", 2018, International Conference on Computational Linguistics and Intelligent Text Processing in Vietnam –H5-Index:19

- Presented submission: Yashothara.S, R.T.Uthayasanker, "A Study on the Utility of Hierarchical Phrase-Based Model For Low Resourced Languages", 2018, International Conference on Computational Linguistics and Intelligent Text Processing in Vietnam -H5-Index:19
- Presented submission: Yashothara.S, R.T.Uthayasanker, G.V.Dias "Pre-processing techniques to improve the translation from Sinhala to Tamil", Asian Language Processing (IALP), 2016 International Conference on. IEEE, 2018
- Presented submission: Yashothara.S, W.S.N.Dilshani, R.T.Uthayasanker, S. Jayasena "Language divergence between Sinhala and Tamil languages", Asian Language Processing (IALP), 2016 International Conference on. IEEE, 2018

1.5.2. Algorithms

This research has produced the following algorithms so far:

- Semi-automatic algorithm for aligning various POS tagsets
- Collocation finding algorithm using PMI.
- Find out the location of the named entity in the parallel corpus.
- Find out the location of POS chunk in the parallel corpus

1.5.3. Software

This research has produced the following algorithms so far:

- Tamil to Sinhala translation in the "Si-Ta" system
- Semi-automatic algorithm for aligning different POS tagsets
- Hierarchical Phrase based machine translation System
- Factored Machine translation system
- Dictionary Tokenizer for SiTa

1.6 Research Methodology

The methodologies of this research are detailed as follows:

- Studying the language divergence between Sinhala and Tamil languages

We discussed the main classes of translation divergences as proposed in [2] with some illustrative examples from Sinhala and Tamil.

- Survey the prior work in POS tagsets of Tamil and Sinhala languages and identify a suitable POS tag set and POS tagger for both languages.

As there are several tagsets available in each language, selections of POS tagset are essential for this study. While choosing a tagset of a language, the usability and standardization are considered. Next chapters describe the identified POS tagsets of Sinhala and Tamil and how the proper tagset is selected to align. Likewise, selection of best automatic POS tagger is also an essential task. The POS tagger which yields high accuracy is selected by comparing different POS tagger.

- Improve and adapt the POS tagger for official documents and evaluate its performance.

POS taggers are created by different people and they focused on different domains. So, there is the need to check the performance of POS taggers in official documents domain and adopting them to our domain.

- Survey on prior work in the morphological analysis of Tamil and Sinhala languages

There are many types of research already held regarding morphological analysis. Different people focused on different levels of morphological analysis. So, we need to come up with the best morphological analysis among all the researches.

- Identify and improve or develop a Tamil morphological analyzer

A proper morphological analyzer with higher accuracy is needed to identify among all available Tamil morphological analyzers. This will help to integrate linguistic features for Statistical Machine Translation system.

- From the analysis of language divergence, come up with a semi-automatic alignment algorithm for aligning multilingual Parts of Speech tagsets

Casting the problem of heterogeneity in POS tagsets as an alignment of two labeled trees and proposed a novel semi-supervised approach algorithm to solve. We plan to evaluate our algorithm using a representative POS tagset chosen from Sinhala and Tamil languages.

- Develop a Hierarchical Phrase-based machine translation system

Hierarchical phrase-based model is to be developed to overcome the issue of word reordering. The parallel texts are to be collected and used to train the hierarchical phase-based model.

- Integrate Tamil POS tagger as well as Sinhala POS tagger (developed under different research) into SiTa system and increase the efficiency.

The bi-lingual sentences are to be created and transformed as factored bi-lingual sentences. Monolingual corpora for Tamil and Sinhala are collected and factored

using identified proper Tamil POS tagger. These sentences will be used for training the factored Statistical machine translation model.

- Pre-processing techniques to improve the translation from Sinhala to Tamil translation

We experimented with few pre-processing techniques based on chunking and segmentation towards addressing challenges which are identified by language divergence. PMI based collocation phrases, POS-based chunks, Named Entities and sub word segments are used to enhance the preprocessing step.

- Evaluate the applicability of System for government reports domain

Evaluating applicability of the system for government reports domain is needed to check the improvement of new system compare to the baseline system. We should come up with improvements and reasons for the improvements also.

1.7 Organization of the Thesis

The thesis is ordered as follows. General introduction of this research and Statistical Machine Translation are presented in chapter 1. Chapter 2 presents the literature survey for available machine translation systems for Tamil and Sinhala languages, existing machine translation using factored model and linguistic tools available for Tamil and Sinhala languages.

Chapter 3 explains the theoretical background of language divergence, semi-automatic alignment of various POS tagsets, Hierarchical phrase-based model, factored model and preprocessing techniques. Chapter 4 explains the methodology details of language divergence, semi-automatic alignment of various POS tagsets, Hierarchical phrase-based model, factored model and preprocessing techniques. How the translation happens using SMT system has been discussed here. This chapter explains how the factored corpora are trained and decoded using SMT Toolkit. Chapter 5 gives the details of implementation of Si-Ta system and tools used in this research. Chapter 6 presents the experiment details of the work presented in the thesis.

Chapter 7 evaluates the work presented in the thesis. It contains subsections for evaluating the translation results in a different scenario, different metrics, and different language pairs. It also describes the training and testing details of SMT toolkit. The output of the developed system is evaluated using BLEU and NIST metrics. Chapter 8 concludes the thesis with a look into future work.

LITERATURE SURVEY

2.1. Overview

This thesis is primarily about improving machine translation between Sinhala and Tamil. We have identified major challenges in the existing systems and proposed several useful insights and techniques to improve the accuracy of machine translation between the above mentioned low resourced languages in the context of official letters. This section reviews the related literature in parts. First, we present various machine translation approaches in the history. Then, we present few machine translation systems in the literature that attempts to translate between Sinhala and Tamil languages. Since the majority of them are SMT and SMT is the rational choice for low resourced languages, we briefly discuss SMT, its challenges and some key ways to tackle them in a low resourced setting. To identify the challenges of traditional SMT, understanding the divergence between languages is an important factor. So we have discussed some existing approaches to identify the divergence between various languages. Based on the results of language divergence, we have attempted to build a semi-automatic algorithm to align different POS tagsets. Prior efforts on POS agreement which are predominantly focused on developing framework on how to standardize POS tagsets of a set of languages are discussed in this chapter. After that, we review various useful Tamil & Sinhala linguistic tools. These tools help in tagsets and factored model alignment.

Factored MT and hierarchical phrase-based MT is useful for morphologically rich low resourced language translations to overcome the challenges such as word reordering, word flow, context-aware word selecting and translating initials and abbreviations in traditional SMT. We present the related literature which utilized hierarchical model and factored model. We have used some preprocessing techniques based on chunking and segmentation to overcome the challenges such as unknown words, context awareness, better word choice, word flow, ambiguity in translation, translating name entities, translating abbreviation and initials and mapping one word with one or more words in the traditional SMT. The literature of those methods used in other languages also mentioned in this chapter.

Section 2.2 discusses about the various machine translation approaches. Details of existing machine translation systems for Tamil and Sinhala languages are discussed in the sections 2.3. More focus has been given to ‘SITA’ translation system

that gives good results for the Official domain of Sinhala and Tamil languages. Section 2.4 describes the previous approaches to identify the divergence between languages. Section 2.5 discusses the literature review about the available POS tagsets for both Tamil and Sinhala languages and linguistic tools such as POS taggers and morphology analyzers for the Sinhala and Tamil languages. Prior efforts of standardizing POS tagsets are described in section 2.6.

Some systems were developed based on hierarchical phrase-based models to overcome the reordering issue. Section 2.7 gives details on hierarchical phrase-based model and the existing hierarchical phrase-based systems. Section 2.8 provides the details of existing systems based on factored model machine translation. There are some language pairs already adopt the factored model for their domains. So this section provides about the factored model and the information about those existing systems. The factored model is used to enhance the performance of traditional statistical machine translation systems. At last the literature survey of the preprocessing techniques is mentioned in section 2.9 and 2.10.

2.2 Various Approaches in Machine Translation

There have been diverse numbers of proposed and implemented approaches to machine translation from the initial stage of using the machine for the process of language translation. The main approaches to machine translation are:

- Rule based or Linguistic approaches
    - Direct approach
    - Interlingua approach
    - Transfer approach
- Non-Linguistic approaches
    - Dictionary based approach
    - Corpus based approach
    - Example based approach
    - Statistical based approach
    - Neural machine translation approach
- Hybrid approach

2.2.1 Rule based or Linguistic approach

Vast linguistic knowledge such as morphological, syntactic and semantic analysis is required for rule-based approaches during the translation. In this approach both knowledge of computer programs and grammar rules are used. It will be supportive in analyzing the text for defining grammatical information and features for words in the source language, translate the word by replacing words by lexicon or same context words in the target language. The principal methodology in machine translation is rule based approach. Rules are written by the use of the linguistic knowledge. These rules will play a vibrant role during different levels of translation.

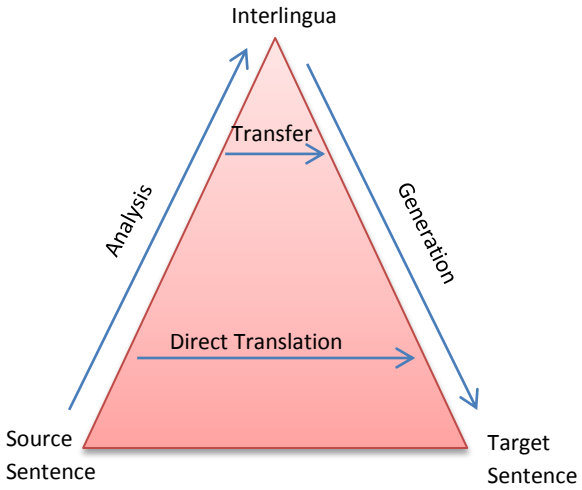


Figure 2.2 Direct, Interlingua and transfer approaches in Machine Translation

Strong examination of the sentence in the terms of syntax and semantic level is the main advantage of this approach. There are difficulties in this method like requirement of massive linguistic knowledge and vast number of rules are needed in order to go through all the features of a language. But it is very difficult to find multilingual experts to come up with grammatically satisfied rules in both languages for translations. Also, the implementation details were very specific for the pair as well as the direction. Therefore implementation of a new pair or new direction requires a larger amount of human work in setting rules. The three different approaches that require linguistic knowledge are direct, Interlingua and transfer based. Figure 1.2 shows the three approaches.

2.2.2 Non-Linguistic Approaches

Any linguistic knowledge is not required explicitly to translate from source language to target language in the non-linguistic approaches. The only resource needed for this

approach is data. Data can be either the dictionaries for the dictionary based approach or bilingual and monolingual corpus for the empirical or corpus-based approaches.

Dictionary-based Approach

A dictionary covering the source and target languages is used in the dictionary based approach. Word level translations are happened in the dictionary based approach. Some pre or post processing steps are required to lemmatize the translated word and include morphological information. Dictionary based approach is very valuable in quickening the human translation by giving same meaning word translation and assisting the human by reducing the efforts of humans to correct the grammar and syntax of a sentence.

Empirical or Corpus-based Approach

Explicit linguistic knowledge is not necessary to the corpus based approaches. A bilingual corpus of both languages and the monolingual corpus of target language are necessary to the execution of this approach. The system is trained using the monolingual and parallel corpus.

Example-Based Approach

Example-Based approach is motivated by repeated translation work where same text with minor variations (only the proper nouns varies in the sentence) needs to be translated several times. The system tries to find the matching sentences/example sentence/phrases in the corpus to match the input text. This involves calculating the closeness of multiple stored source sentences to match the given text. Then the corresponding target sentences are combined to generate the translation output. There are steps such as example acquistion, example base and mangement, example application and synthesis. EBMT system could produce novel sentences and not just reproduce previous sentences. Matching, alignment and recombination are the three steps in EBMT system.

Statistical approach

Statistical machine translation approach is one of the corpus-based machine translation approaches. It is based on the statistical models that are made by analyzing the parallel corpus and monolingual corpus. The original idea of SMT was initiated by

Brown et al [3] based on the Bayes Theorem. Basically, two probabilistic models are being used; Translation model and Language model. The output is generated by maximizing the conditional probability for the target given the source language. Figure 1.3 shows the simplified block diagram of SMT system.

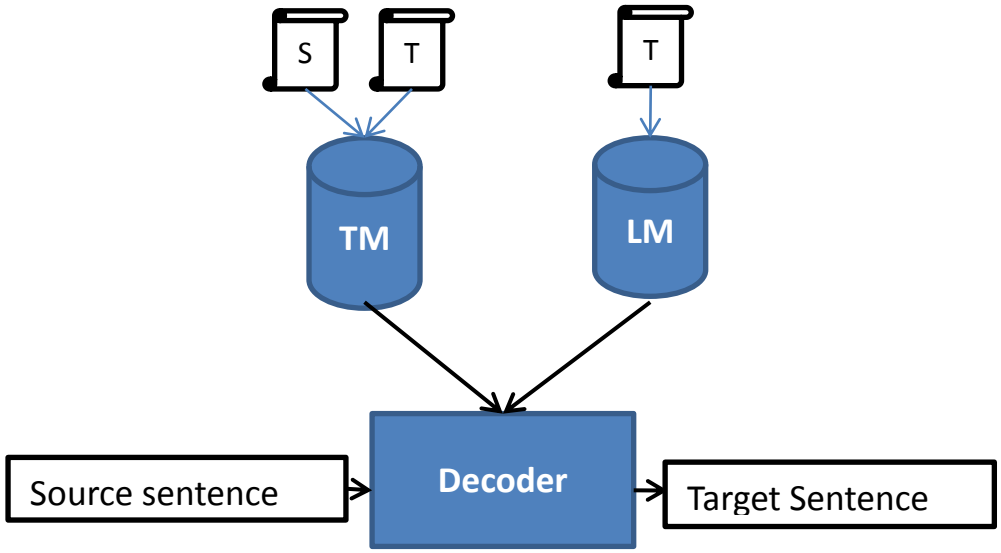


Figure 2.3 Block diagram of SMT system. S=Source language sentences, T=Target language sentences, TM=Translation model, LM=Language model

The advantages of statistical approach over other machine translation approaches are as follows:

- Manually translated aligned parallel texts of language pairs, books accessible in both languages can be used in the statistical approach. Machine readable texts can be properly used for this approach.
- SMT systems are language independent that means any language pair can be adapted to the system if we have fair amount of corpus.
- As there is a high investment in creating manual linguistic rules and that rules are specific to particular language pair, rule-based machine translation systems are generally costly, whereas SMT systems can be adapted for any pair of languages if bilingual corpora for that particular language pair is available.
- More acceptable translations are given by SMT system compared to other systems.

Neural Machine Translation System

Recently neural approaches based on deep learning techniques for machine translation has have shown the promising result for many language pairs over the statistical machine translation. This methodology does the full translation process with the single

neural model. The commonly used approach is 'encoder-decoder' framework [4]. Here the source sentence is encoded into a vector which is called context vector. Then in the decoder process, the translation is generated to this vector. Since the translation process happens for the whole sentences at one step rather than segments, the fluency of the output has been increased than SMT approach. Yet, the main shortcoming of NMT is that it falls back to the unknown words. The quality is confirmed for closed vocabulary. Therefore the system works inferior for low-resource setups while SMT output was much better. But for high resource language, it performs better compared to SMT.

2.2.3 Hybrid machine translation system

Benefits of both statistical and rule based approaches are adapted in Hybrid machine translation approach. Hybrid approach is used in commercial translation systems such as Asia Online and Systran. Hybrid machine translation approaches differ in many aspects: Rule-based system with post-processing by the statistical approach and statistical machine translation system with pre-processing by the rule-based approach.

2.3. Existing machine translation systems for Sinhala and Tamil languages

Considering local languages of Sri Lanka (Sinhala and Tamil) very minimal numbers of researches have been carried out to date. Sinhala and Tamil languages lack in necessary natural language resources and tools hence classified as low resourced languages. This limits the success achievable in machine translation to and from those languages. Tamil and Sinhala languages which gain importance since both of them are acknowledged as official languages of Sri Lanka. Further, since these two languages are considered low resourced languages, these efforts gain more importance. The Sinhala language belongs to the Indo Aryan language family and the Tamil language belongs to the Dravidian family. As these languages have been in contact for a long period of time, they share notable resemblances in morphology and syntax. This makes it sensible to translate between them. Even though a small amount of parallel data is available, some notable amount of effort is needed to translate between Tamil and Sinhala languages. The data carried out in those researches have used news articles with marginal amounts of parliament order papers [5] [6] [7] [8] [9].

As a first attempt, Ruvan Weerasinghe proposed a basic SMT approach to Sinhala and Tamil languages [5]. Same Sinhala corpus and its translated Tamil version corpus are used in the learning process of Statistical Machine Translation between Tamil and Sinhala in this work. The CMU-Cambridge Toolkit [10] was used to build the language model from a target language monolingual plain text corpus which is based on n-gram statistics. GIZA++ system is used to build Translation model using parallel data of both languages. ISI-Rewrite decoder with various different parameters and smoothing methods was used to decoding process to come up with a best possible scheme for the Sinhala and Tamil language pair. The evaluation of the output produced by the system was based on BLEU [11] score. A set of WSWS articles [12] available on the site (www.wsws.org) during 2002, which contained translated articles in the English, Sinhala and Tamil languages, were chosen to form a small tri-lingual corpus and used in this research. The manually aligned parallel corpus had 4064 sentences of Sinhala and Tamil corpora. A set of 162 Tamil test sentences taken from the same website was used to evaluate the system. After testing with multiple translation models, they have achieved a best BLEU score of 0.1362 for this task [5]. In this work, they have accessed only to a single translation which was taken to be the reference translation. Here the domain of the research is on news articles and they could only collect few amounts of parallel data. Due to the lack of linguistic tools such as lemmatizers, taggers etc. for Sinhala and Tamil, all language processing done used raw words and were based on statistical information. So they didn't use any linguistic information is used in their approach.

Following that work, S. Sripirakas et al. [6] proposed translation system between Tamil and Sinhala languages. They used a parallel corpus based on parliament order papers which were obtained from UCSC-LTRL [13]. They demonstrate only the preliminary system which runs both directions of Tamil and Sinhala languages [17]. System favored only to parliament order papers domain specific translations, caused by the nature of prepared corpus. GIZA++ was used to build translation model and SRILM [14] was used to build a language model. Moses toolkit was used to decode. MERT [15] Module, TER module, NIST module were also the fundamental components of this built-in the system. The evaluation is based on BLEU, NIST and TER metrics. 5697 parallel sentences were used to build the translation model. Language model contains 6566 sentences in Sinhala language and

75051 sentences in Tamil language which are monolingual corpus [6]. For tuning and testing, a different set of 200 sentences were used. They got 0.4277 as BLEU score in Sinhala to Tamil direction and 0.5599 in Tamil to Sinhala direction. They have evaluated the system with and without MERT tuning. From the experiments they have concluded that these metrics favor the Tamil-to-Sinhala translation than the Sinhala-to-Tamil [6]. Here also domain of the research is on government order papers and they have collected only 5000+ sentences of parallel data. So this research also did not use any linguistic information.

Some essential factors to consider for building SMT between Sinhala and Tamil languages have been identified in Sakthithasan el al work [16]. The effort is hard to generalize because of the limited amount of data and the restricted domain. Another study [17], discovered the applicability of the Kernel Ridge Regression technique in the translation of Sinhala to Tamil direction. This research occasioned in a hybrid of traditional phrase-based SMT and Kernel Ridge Regression with two novel solutions for the pre-image problem.

Pushpananda et al [7] investigate on the behavior of SMT systems against the size of data for the parallel corpus. MOSES toolkit with GIZA++ was used in standard alignment. Tri-gram language models were trained for building language model on the target side of the parallel data by using SRILM tool. The score is based on BLEU. Language model contained 850,000 sentences in Sinhala side and 407,578 in Tamil side. Both these are open domain corpora mainly with newspaper articles and Technical writing. Sinhala-Tamil Parallel Corpus consisted of 25500 parallel sentences. This parallel corpus was also open domain including mainly newspaper texts and technical writing. 500 sentences were used as tuning dataset [7]. They calculate BLEU score from 5000 to 25000 sentences in both directions. Through that, they have noticed that BLEU score increases according to amount of parallel data. As highest BLEU score they have achieved around 13 in Tamil to Sinhala direction and around 10 in Sinhala to Tamil direction [7]. Here as their parallel corpora increased they achieved better BLEU score. But they did not use any linguistic information.

Rajpirathap et al [8] focused on the research to develop a real-time communication system which can perform SMT for Tamil and Sinhala. The parallel data was taken from parliament order papers on budget proceedings. They have used

over 5000 phrases from each language to train the system. 6550 sentences in Sinhala side and 6104 sentences in Tamil side were used to build a language model. Language model was built by IRSTLM considering 2-gram count. Translation model is implemented by GIZA++ tool. BLEU and NIST were used as evaluation metrics score. They have achieved 0.6693in Tamil to Sinhala direction and around 0.5957 in Sinhala to Tamil direction with MERT tuning [8]. They have demonstrated an analysis of the system behavior with and without MERT tuning of the weights for different models and features (LM, TM, word alignment, lexical reordering). And they have compared the average consumed the time of the translation in both directions. Here also translation is domain specific and did not use any linguistic information.

Pushpananda et al. [9] extend the work of their own [7] where it uses the same data set as of the previous work to elaborate a study on incorporating an unsupervised morphological analyzer to the system using the Morfessor algorithm [18]. Morfessor algorithm was used to find morpheme-like units of the source and target languages in order to build the translation and language models. Three sets of experiments such as with a word based (Traditional SMT system), fully morpheme-like and semi morpheme-like segmentation systems for the Sinhala-Tamil language pair are done by them. Moses toolkit [19] along with GIZA++ was used to build the traditional SMT system. The fully morpheme-like system used morpheme-like units as the smallest unit and phrase-based SMT modeling approach was used similarly to the baseline system. Semi morpheme-like system has combined all the prefixes and stems together and separately merged the suffixes. In this research, parallel sentences contained written and spoken languages. They have done experiments only Tamil to Sinhala direction. BLEU score is calculated based on word-based, fully segmented and semi segmented approaches. It was clearly indicated that the word-based baseline system gives better BLEU score results compare to other settings. However, BLEU score value increased, when they increased the language model size up to 7-gram [9]. The results reveal that the system significantly reduces the OOV problem. Even though they have used linguistic information, our approach is different from them as we used POS tagged data and different preprocessing techniques.

Recently a research has been carried out on the development of Sinhala-to-Tamil and vice versa SMT system for official government letters which are

“unpublished” [20] [21]. They have conducted experiments using a test set (Test-1) that was randomly picked from the collection of letters from where the training and tuning data are also derived and a test set (Test-2) from a different set of letters, from which no data was included in training or tuning. This system was developed with emphasis given to domain adaptation. They concluded that Test-1 gives better results than Test-2 due to out of vocabulary. They had two domains of data such as in-domain and out-domain. In-domain contains official letters from government department. Meanwhile, the size of in-domain data was lesser, additional data was collected from other government sources such as annual reports, parliament order papers, circulars, and establishment codes. Totally 22,073 sentences were in the parallel corpus. Moses tool kit with GIZA++ was used to build the translation model and SRILM was used to build the language model. They have conducted the experiments using five scenarios using BLEU score metrics [21]. They got a better score as 25.05 by using baseline system and integrating pseudo-in-domain data in Language Model and Translation Model for Sinhala to Tamil direction. They got a better score as 32.85 by using baseline system and integrating pseudo-in-domain data in Translation Model for Tamil to Sinhala direction [21]. In this approach also they did not consider adding linguistic information. So as extending of this project, I am planning to add preprocessing techniques and linguistic information to improve the translation between these languages.

Except for the SMT system, Pasindu et al. [22] focused on building domain-specific Neural Machine Translation Systems for Tamil and Sinhala languages. They came up with the novel approach of using word phrases to enhance domain specific NMT translation. And also they empirically tested the applicability of monolingual corpora of the target language. The domain of this translation was official government documents of Sri Lanka. Parallel corpus was collected from above mentioned Farhath et al.’s “unpublished” [20] work. Then Open Source NMT system openNMT [23] was used for the experiments. BLEU score matrix was used to evaluate the quality of the translation. 23611 parallel sentences were used in this approach. They trained separate models for both directions of translation by adding 5000 more-word phrases to the early training dataset each time. Experiments were carried out for 5k, 10k, 15k, 20k, 25k, 30k, 35k, 40k, 45k and 47k number of word phrases [22]. They got 7.50 for Sinhala to Tamil translation and 12.75 for their Tamil to Sinhala translation systems as

their highest score. As the parallel corpus is small, NMT system did not get good result comparing to SMT system. And here also any linguistic information was not used.

There is no published literature on applying factored phrase-based SMT between Tamil and Sinhala languages. But, traditional statistical based machine translation (SMT) mostly fails to produce quality output for long sentences. Out of all these systems, the best BLEU score for Sinhala-to-Tamil translation was 37.01 and for Tamil-to-Sinhala translation was 46.64 [21]. The available systems, focused domain, parallel corpora details and results are described in Table 2.1.

Table 2.1 Details and comparison of existing Sinhala-Tamil translation systems in terms of corpus size, domain, accuracy and linguistic information

	Parallel Sentences	**Language model Sentences**		**Test Sentences**	**No of Words (Training)**		**Domain**	**BLEU Score**	**Linguistic Information Added**
		Sinhala	Tamil		**Sinhala**	**Tamil**			
R.Weerasinghe	4064	4064	4064	162	65k	46k	News Article	0.1362	No
S.Sripirakas et al.	5697	6566	75051	200	99457	72393	Parliament order papers	0.4277 (Sin to Tam) 0.5599 (Tam to Sin)	No
Pushpananda et al.	25500	850,000	407,578	2500	252,101	219,017	Newspaper article, technical writing	10(Sin to Tam) 13(Tam to Sin)	No
Rajpirathap et al	5000	6550	6104	200	~100,000	~100,000	Parliament order paper	0.5957 (Sin to Tam) 0.6693 (Tam to Sin)	No
Pushpanda et al.	25,500	850,000	-	162	252,101	219,017	Magazines, books, articles	12.99(Tam-Sin)	Yes
Farhath et al.	22,073	4,917,149	1,623,768	340	318,347	270337	Government official letters, annual	37.01(Sin-Tam) 46.64 (Tam-	No

							reports, parliament order papers, circulars, and establishment codes	Sin)	
Pasindu et al.	23611	10000	10000	500	346030	293821	Government Official documents	7.50(Sin-Tam) 12.75(Tam-Sin)	No

To overcome challenges such as reordering, abbreviations and initials, word flow of the sentence, data sparseness and one word can map with one or more word, we have improved the SMT using Hierarchical phrase-based model, Factored model with POS, Segmentation, and Chunking techniques.

2.4 Existing Approaches in Language Divergence

Divergence is one of the common obstacles in the machine translation systems. It is essential to identify the different types of divergences to get correct translation. In the literature of machine translation, some efforts have been carried out to classify the types of translation divergence between a different pair of natural languages. This section describes those efforts.

Dorr's solutions for the divergence [24] were an outcome of such an initial blow to create classification rules for language divergence. They demonstrated a systematic solution to the divergence issue can be derived from the formalization of 2 types such as the linguistically grounded classes upon which lexical-semantic divergences are based and the techniques by which lexical-semantic divergences are resolved. They came up with 7 categories in the classification of language divergence and solutions to overcome these issues. This became the foundation for several other researches ([25], [26], [27], [28], [29]) in language divergence classification.

Saboor and Khan [25] focused on the lexical semantic divergence between Urdu and English languages. They have identified six distinct types and generalizations are made on the basis of examples. They came up with efficient examples from the parallel corpus which are helpful in the alignment and recombination stages of EBMT to give good quality translation. They proposed an

algorithm for the identification of lexical-semantic divergence. All strategies made from the examples are applied on the bilingual corpus. To keep the differentiation between normal examples and diverged examples they suggested tagging them with <DIV>. This identification will benefit in the adaptation and recombination stage of EBMT and will help ineffectual translation of input sentences of source sentence into equivalent target sentence.

Dash [27] discussed some of the major divergences that observed in English to Bengali translation when the source sentence is realized differently in the target language. They have interrogated how the distinct linguistic and extra-linguistic constraints can perform decisive roles in divergences and other issues. They have classified syntactic divergence into three categories. Lexical semantic divergence is classified into seven categories.

Behera et al. [29] presented different types of linguistic divergences: the lexical-semantic and syntactic. This study assists in identifying and resolving the divergent features between English and Bhojpuri language pair. They have followed Dorr's theoretical framework in the classification and resolution procedure. Additionally, so far as the methodology is concerned, they have followed to the Dorr's Lexical Conceptual Structure for the resolution of divergences.

Harold Dharmasenan Thampoe [30] has studied on the convergence patterns based on the morphosyntactic features of modern spoken Sinhala and Tamil languages. The study was focused to find out the features shared and not shared by both languages, the reasons for sharing similar features and the morphosyntactic restructuring of Sinhala on the model of Tamil. The morphosyntactic features of the two languages have analyzed at macro- and micro-levels. At the macro-level, a wide range of morphosyntactic features of Tamil and Sinhala, and those of seven other languages of the region are compared with a view to determining the origins of these features and showing the large-scale morphosyntactic convergence between Sinhala and Tamil and the divergence between Sinhala and other languages. At the micro-level two morphosyntactic phenomena, namely, null arguments and focus constructions have studied. Accordingly, the findings prove that most of the similarities are undergone in spoken Sinhala language with Tamil language due to language contact.

But there are no efforts focusing on the divergence between Tamil and Sinhala languages, even though there are some automatic machine translators. Without identifying divergence between these languages, it is difficult to come up with challenges and solutions in the automatic machine translation systems. So, this research focuses on the divergence between Sinhala and Tamil languages to assist the machine translation.

2.5 Literature review about the available POS tagsets and linguistic tools for both Tamil and Sinhala languages

Parts of Speech (POS) is a category to which a word is assigned in conformity with its morphosyntactic functions [31]. Examples of parts of speech are noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection. It is alike to the tokenizing process tokenization for computer languages. The process of assigning the POS label to words in a given text is an important aspect of natural language processing. The initial task of any POS tagging process is to choose various POS tags which are word classes such as noun, verb, adjective, etc in a language. POS tagging is considered as an important process in speech recognition to identify the correct pronunciation, natural language parsing, morphological parsing, information retrieval and machine translation. The importance of POS tagging inspired various researchers to work independently in developing POS tagsets for a language. This limited the reusability of tagged corpus among NLP researchers of the same language. But the main challenges in POS tagging are solving the complexity and ambiguity of words.

POS tagger is a piece of software that reads the text in some language and assigns parts of speech to each word such as noun, verb, adjective, etc [33]. Different approaches were used for Part-of-Speech (POS) tagging such as rule-based, stochastic, and transformation-based learning approaches [33]. Set of hand-written rules are used to assign a tag to each word in Rule-based taggers. A training corpus is used to pick up the most probable for word in the Stochastic/Probabilistic approach. Transformation approach combines the rule-based and Stochastic approaches. Like stochastic approach, it picks up the most likely tag based on a training corpus and then put on a certain set of rules to see whether the tag should be changed to anything else. For further use, new rules learned from the process are saved.

2.5.1 Tamil Language POS Tagsets

There are several tagsets available in Tamil language, selection of a POS tagset is essential for this study. While choosing a tagset of a language, the usability and standardization are considered. This subsection describes the existing POS tagsets of Tamil language. For the Tamil language, there are plenty of tagsets. We considered nine tagsets ([34], [35], [33], [36], [37], [38], [39], [40], [41]) before choosing an appropriate one for this study.

Amrita POS tagset

The customized POS tagset has developed by and Dhanalakshmi V and et al. [41] which contains 32 tags without considering the inflections. 32 tags are used in their approach to minimize the complexity of tagging process. Because more tags with grammatical features cause splitting each inflected word into a base form. Compound tags were used for compound nouns (NNC) and compound proper nouns (NNPC). Different from other approaches, Tag VBG is used to tag verbal nouns and participle nouns.

Madhu Ramanathan Tagset

Madhu Ramanathan and et al. [36] have used 12 tags for tagging purpose. These tags were chosen because of the frequent occurrences and also appear in other languages (Hindi, English French). The chosen twelve tags are Noun, Compound noun, Pronoun, Compound Pronoun, Verb, Adjective, Adverb, Conjunction, Preposition, Number, Others and Punctuation marks. They did not go further levels of tag sets even those information is useful in tagging.

IIIT Tagset

This tagset was developed by IIIT [37], Hyderabad. Basic Penn Treebank tagset is used in this work by modifying some of the basic tags and bringing some tags to address the special feature of Indian languages. Tags are decided on grainy linguistic information with an idea to develop it to finer knowledge if required. The annotation standards for POS tagging for Indian languages include 26 tags. These tags are not only for Tamil but also for all Indian languages. Postposition, Quantifiers, Quantifiers Number, Verb Finite Main, Verb Non-Finite Adjectival, Verb Non-finite Adverbial, Verb Non-Finite Nominal and Question Words are the modified tags from Penn Tree tagset. Noun Location, Intensifier, Negative, Compound Nouns and

Compound Proper Nouns are the extra tags included in their work compared to Penn Tree Tagset. It is a flat tag set.

MSRI tagset

This tagset is developed MSRI (Microsoft Research India Pvt Ltd) in 2008. The researchers aim to provide a comprehensive tagset which captures as much information as possible from tagging. There are 9 tags in this tag set such as Nouns, Pronouns, Verbs, Nominal Modifier, Demonstrative, Adverb, Particle, Punctuation and Residual [33]. It has been further divided into 14 categories.

Lakshmana Pandian and Geetha Tagset

This work focuses on multi-level morphology in determining the POS category of a word [38]. The stem, the pre-final and final morpheme components committed to the word that is the words derivative form normally contribute to choosing the POS category of the word. Context and combinations of morphology components also influence the POS category of the word. There are 35 POS tags in this tagset.

Selvam and Natarajan Tagset

They have focused on morphological features such as case suffixes (Accusative, Dative, Instrumental, Sociative and etc.) used with a noun, number and gender variations for noun, verbal suffixes according to tense, person and other suffixes and tense and negative variations of adjectives and prepositions [39]. According to above morphology features they have come up with more than 600 tagsets.

CIIL Tagset

CIIL (Central Institute of Indian Languages) Mysore has developed this tag set. There are 71 tags for Tamil. According to the inflation in noun and verb, the number of tags will increase. It has 30 noun forms including pronoun categories and 25 verb forms including participle forms [40].

LDC-IL Tagset

It is a multi-level tagset. So according to the purpose, we could select the proper number of tags. Top level tags are further divided to get the bottom level tags. In bottom layer morph syntactic features also covered. LDC-IL has 13 top-level categories. These tags further divide into some subcategories according to the level [35].

BIS Tagset

Bureau of Indian Standards (BIS) is recommended as common tagset for POS annotation of Indian languages. Many tags in BIS are same as LDC-IL tagset. It groups together unknown, punctuation and residual into one tag. Except for adjective, adverb, and postposition tags, all other tags have some two or more sub-categories. There are three levels in this tagset. It has 11 tags in level I such as Noun, Pronoun, Demonstrative, Verb, Adjective, Adverb, Postposition, Conjunction, Particles, Quantifiers and Residuals and 32 in Level II tags. Level II is made by further subdividing the level I tags [34].

2.5.2 Tamil Language POS Tagger

For Tamil language, various methodologies are used for POS Tagging. There are POS taggers for Tamil language using different approaches. We considered nine taggers ([38], [42], [39], [43], [44], [45], [46]) before choosing an appropriate one for this study. Arulmozhi et al. developed a POS tagger for Tamil using rule-based approach [47]. This POS tagger gave only the major tags for a word and the subtags are overlooked during evaluation. A hybrid POS tagger for Tamil using HMM technique and a rule-based system was also developed [48]. A POS tagger based on phonological approach was proposed by Vasu Ranganathan. Ganesan proposed a POS tagger and applied on CIIL corpus. Another rule-based POS tagger was developed by M.Selvam and A.M.Natarajan in 2009. Dhanalakshmi Vet al developed two taggers and own tagset.

Vasu Ranganathan's Tagtamil (2001)

Vasu Renganathan developed a POS tagger "Tagtamil" which is based on Lexical phonological approach. Using index method, Tagtamil does morphotactics of morphological processing of verbs. It handles both tagging and generation [43].

Ganesan's POS tagger (2007)

Ganesan proposed a POS tagger and applied on CIIL corpus. It works efficiency in CIIL corpus. He created own tagset for his tagger. The tagset used in the tagger is very rich in morphology. He tagged a portion of CIIL corpus by using a dictionary as well as a morphological analyzer. A manual correction was done and the rest of the corpus was trained. The tags are added morpheme by morpheme. They did not test the efficiency in another corpus [44].

Kathambam of RCILTS-Tamil

Kathambam uses heuristic rules based on Tamil linguistics for tagging and without either using the dictionary or the morphological analyzer. It yields 80% efficiency for large documents. Twelve heuristic rules were used and the tags were identified based on PNG, tense and case markers. Standalone words are checked with the lists stored in the tagger. "Fill in the rule" was used to tag unknown words using bigram approach from previous word category [45].

Lakshmana Pandian and Geetha POS Tagger

Lakshmana Pandian and Geetha [38] developed a Parts of Speech tagger and chunker using CRF Models. Maximum Entropy Markov Models (MEMMs) and other discriminative Markov models were avoided in this approach due to the limitations in those methods.

Selvam and Natarajan POS tagger

A rule-based Morphological analyzer and POS tagger developed by Selvam and Natarajan [39]. The system was improved Projection and induction techniques. They have used well defined morphological rules to build morphological analyzer and POS tagger. Adopting alignment-projection techniques and categorical information, a well-formed POS tagged sentences in Tamil were attained for the Bible corpus. For

projecting POS tags and alignment and lemmatization, they applied alignment and projection techniques. Morphological induction techniques were used for inducing root words from English to Tamil. The generated tagset contained 600 POS tags. They got an improved accuracy of about 85.56% [39].

Dhanalakshmi V et al. **SVM POS tagger**

Dhanalakshmi et al. [41] developed a POS tagger based on Linear Programming approach [42]. For the POS tagger, they have used their own tagset contains 32 tags. This tagger is based on SVM methodology based on linear programming. They have used a corpus of 25,000 sentences to train the system. The testing data contained 10,000 sentences. They got best overall accuracy of 95.63%.

Dhanalakshmi V et al. **machine learning POS tagger**

Dhanalakshmi et al. [42] developed another POS tagger using machine learning techniques. Linguistic knowledge is automatically extracted from the training annotated corpora. The tagset used to develop SVM tagger is used in this POS tagger also. Two hundred and twenty-five thousand words were used to train the system. Support vector machine algorithms were used to train and test the POS tagger system. They stated accuracy as 95.64%.

AUKBC POS tagger

Anna University developed a POS tagger. They have used BIS tagset in their POS tagger. This tagset is standardized by the Government of India and Government of Tamil Nadu. This takes a text file in UTF-8 as input and assigns the part of speech tag (e.g. noun, verb, adjective etc.) to each word in the sentence. This uses a Machine Learning based approach. We have used Conditional Random Fields (CRFs), to develop the tagging engine. The engine is trained using a 500K word corpus of a Tamil Novel titled "Ponniyan Selvan" written by "Kalki Krishnamurthy". The language style used here contemporary Tamil which is in use currently. Thus this could be used for any general text. The engine has been evaluated by performing 10-fold experiments. This has an accuracy of 95.42%. The most common type of errors are Proper Noun (N_NNP) being tagged as Common Noun (N_NN) and vice versa, Relative Participle (RP) being tagged as Adjective and Verbal Nouns as Nouns [49], [46].

2.5.3 Sinhala Language POS Tagsets

There are two tagsets available for the Sinhala language such as University of Colombo School of Computing (UCSC) tagset which was developed by University of Colombo [50] and UOM tagset by University of Moratuwa [51]. The details of the tagsets are described in the next subsections.

UCSC Tag set for Sinhala Language

This tagset is designed by University of Colombo, Sri Lanka. UCSC tagset contains 29 tags which includes foreign words and symbols. There are 3 versions in UCSC tagset. In the version 3 tagset, they have included a Common Noun root tag and further split the Verb-particle tag. Separation of Particles and Postpositions and Separation of Compound nous has been considered when defining tagset [50]. This tagset has two major limitations. The major limitation is some Sinhala words which do not belong within 27 tags assign under the unknown tag. Some examples are හැකි - hæki "can",යුතු -yuthu "should/must", නොහැකි - nohæki "cannot", කුම - kumana "which", ඉටු - itu, සිදු - sidu,පත් -path and බව - bava [51]. But those words have some special linguistics features and can be grouped into new tag category. The second limitation is inflection based grammatical variations of words have not been taken in this tagset. For example, common nouns in Sinhala that get inflected based on cases (Nominative: පොත -potha "the book", Accusative: පොතක් -pothak "a book", Dative: පොතට -pothata "to the book", Genitive: පොතේ/පොතෙහි -pothe/pothehi "in the book", Instrumental: පොතෙන් - pothen "from the book") are tagged under a single tag [50].

POS tagset designed by University of Moratuwa

This tagset has been improved from the UCSC tagset by overcoming the limitations. Some comparable tagsets are borrowed from the Penn Treebank tagset. The tagset is divided into three levels. In each level, tags are divided further based on inflecting factors or contextual definitions. In the third level, there are 148 tags [51]. The tag set is organized in a hierarchical manner.

- **Level I Tags**

It only contains the primary top-level part of speech such as Nouns (නාම - nāma), Adjectives (නාම විශේෂණ - nāma viśēṣaṇa), Verbs (ක්‍රියා-kriya), Adverbs (ක්‍රියා විශේෂණ- kriya viśēṣaṇa), and Nipāta (නිපාත) [51].

- **Level II Tags**

Level I tagset is further divided to get the level II tagset. Nouns are divided into seven further categories. Adjectives are further categorized into three. Verbs are divided into five sub categories. Nipātha is further divided into 8 categories. There are some additional six tags added in Level II tag set. So totally there are 29 tags in this level.

- **Level III Tags**

Based on the number, gender, person, animacy, definiteness, case, and tense, noun and verb can be inflected. At the most fine-grained level, this tag set contains a total of 148 tags [51].

We did not go further to Level III as it will not give the best results according to the expectation in NLP applications. The main negative in the majority of tagsets is that they take the verb and noun inflections into consideration for tagging. Hence at the tagging time, one needs to split each and every inflected word into morphemes in the corpus. It is a tough and time-consuming process. At POS level, one needs to determine only the word's grammatical category, which can be done using a limited number of tag set. The inflectional forms can be taken care by morph analyzer. Moreover, a large number of tags will lead to more complexity which in turn reduces the tagging accuracy.

2.4.4 Sinhala Language POS Taggers

For Sinhala language also, various methodologies have been used for POS Tagging. For Sinhala language, there was four reported work for implementing a POS tagger. A Hidden Markov Model (HMM) ased POS tagger was developed using bigram model with 60% of accuracy. Another HMM-based approach was proposed with a 62% of accuracy. A hybrid approach based on bi-gram HMM and rule-based proposed in 2016 with 72% accuracy. Available POS taggers are described in this section.

Jayaweera et al. POS tagger

They presented a POS tagger for Sinhala language using Hidden Markov Model (HMM). The inputs of this tagger are a sentence, a tagset and tagged corpus. The output of the system is tagged sentence. By counting the tag sequence probability P(ti|ti-1) and a word-likelihood probability P(wi|t) from the given annotated corpora, the tagging process is done. Here from the annotated corpora, linguistic knowledge is extracted automatically [52]. UCSC/LRTL (2005) tagset and corpora were used in this

research. The current tagset consists of 29 morphological syntactic tags. The tagger gives accuracy more than 80%.

Gunasekara et al. POS tagger

They proposed a hybrid POS tagger by combining the knowledge of rule-based and stochastic tagging approaches. Initially, they have built a Hidden Markov model-based stochastic tagger based on bi-gram probabilities [50]. They used a stemmer in the tagging process to enhance the accuracy of the tagger. They have experimented with different POS tagsets and came up with best tagset. Since Sinhala is a morphologically rich language, for the words which are not in the training set, they have used rules based on morphological features. Further, an experiment is carried out to find out whether the implemented hybrid tagger can be used to enhance the size of the dataset. The tagger achieved an overall accuracy of 72% when the average unknown word percentage is 20% [50].

M. Jeyasurya et al. POS tagger

They proposed a Hidden Markov model POS tagger for the Sinhala language. Lexical items with multiple POS tags are handled in this approach. This POS tagger can predict the POS tags of the previously unseen word. They used a stochastic approach with Hidden Markov Model (HMM) with tri-gram probabilities in the training and tagging model. The tagger learns the lexical items and the tri-gram probabilities using a POS tag annotated corpus. The tagger achieved an overall accuracy of 62%. About 24% errors were for words which belong to the unknown category in the training corpus [53]. The lack of a Named Entity recognizer has also contributed to 10% of the overall error.

UOM POS tagger

Sandareka et al. [51] proposed a new multi-level POS tagset and POS tagger based on Support Vector Machine for the Sinhala language. They have created new tagset to overcome the already available tagsets. The accuracy of available Sinhala Part-Of-Speech taggers, which are based on Hidden Markov Models, still falls far behind state of the art. Researchers reported an overall accuracy of 84.68% with 59.86% accuracy for unknown words and 87.12% for known words when the test set contains 10% of unknown words.

2.6 Existing Approach for Alignment between Different POS Tagsets

Prior efforts on POS agreement predominantly focused on either developing framework about how to standardize POS tagsets of a set of languages and using the guidelines of POS standardization to create a new standardized tagset or mapping from different tree-bank tagsets to universal set. Below, we present the literature review of both approaches.

2.6.1 Existing Approaches on POS Standardization

There are several POS standardization efforts carried out by NLP researchers around the world. EAGLES guidelines [54] were an outcome of such an initial blow to create standards that are common across languages. The EAGLES Guidelines yield governance for analytic information about the language of a text, particularly for identifying morphosyntactic and syntactic features relevant in computational linguistics. The aim of these EAGLES guidelines is interchangeability and reusability of annotated corpora in different languages. According to the morphologic features, top level is further divided. Here further diving rules are optional. In this approach, they did not create newly standardized tagset using the guidelines they gave. This became the foundation for several other types of research ([55], [56], [57], [58]) in leveraging morphosyntactic and syntactic features to develop common standards across multiple languages. The main weakness to the EAGLES guidelines is that they cover only a small fraction of the world's nine languages such as English, Dutch, German, Danish, French, Spanish, Portuguese, Italian and Greek.

LE-PAROLE project [56] formed a multilingual corpus for fourteen European languages; morphosyntactically annotated according to a common core PAROLE tagset, extended with a set of language-specific features. MULTEXT [8] focused on tools, corpora and linguistic features for multi-languages, with the extension of other languages. But this project also mostly focuses on European languages to make the standardization among them. However, a spin-off MULTEXT-EAST [57] gradually added morphosyntactic descriptions of sixteen languages, including Persian or Uralic languages. The MULTEXT-EAST dataset embodies the EAGLES-based morphosyntactic specifications, morphosyntactic lexicons, and annotated multilingual corpora. General mechanisms for lexical specification has been contributed by MULTEXT-EAST, and it has provided a test of the extensibility of standards and

tools beyond the languages for which they were originally developed.

Early works on POS standardization were predominantly in European languages. One of the early works on standardizing Indian languages was by, Baskaran et al. [55] who have focused on designing a common POS tagset framework for eight Indian languages by considering equivalent morphosyntactic phenomena consistently across all languages. They have designed a common tagset framework for Indian languages using the EAGLES guidelines as a model. Hierarchical and decomposable tagsets were used in the framework as it is a recognized method for creating a common tagset framework for multiple languages [55]. They focused only on the morphosyntactic aspects of the Indian languages for encoding in the framework assuming the existence of morphological analyzers and choice of granularity left on users' side. They have created 3 levels in their framework and the top level was with 12 categories.

The BIS has released Unified Parts of Speech (POS) Standard in Indian Languages with the consideration of morphologic syntactic features of Indian languages. According to the morphological features, the top level is subdivided into next two levels [34]. This POS schema relies on W3C XML Internalization best practices, ISO 639-3 Language Codes for Language Identification, ISO 12620:1999 as metadata definition. One to one mapping table for all the labels is used in POS Schema. They have covered 22 Indian languages in their work.

Nitish Chandra et al. [35] claimed that the tagset for which taggers perform best should be the standard tagset to be followed, and sought for the POS tagset which yields the highest accuracy during the automatic POS tagging for a set of Indian languages [35]. Unlike prior efforts, designing a new common framework was not the focus of Nitish Chandra et al [35]. They have done experiments by identifying standard tagsets such as IIIT (ILMT) tagset, BIS tagset LDC-IL tagset, AU-KBC tagset, MSRI-Sanskrit tagset and CIIL Mysore tag set for Indian languages. After that, they have measured the performance in tagging by using different POS tagset. Performance measurement based on the ratio between correctly tagged words and words tagged. They calculated Precision and Recall to Hindi, Bengali, Telugu and Tamil languages.

POS standardization focuses on designing a common tagset framework that can exploit similarity. Mapping from existing tagset to the standardized tagset was not considered in the above approaches. But there are some on mapping from different tree-bank tagsets to the universal tagset.

2.6.2 Existing Approaches on Mapping From Different Tree-Bank Tagsets To Universal Set

Instead of standardizing morphosyntactic tagging, there are some efforts of mapping existing tagsets to universal tagset which they created. A Universal Part-of-Speech Tagset was proposed by McDonald et al. [31]. The tagset consists of twelve universal part-of-speech categories. In addition to the tagset, they evolved a mapping from 25 different tree-bank tagsets to this universal set. As a result, this universal tagset and mapping generated a dataset consisted of common parts-of-speech for 22 different languages. When corpora with common tagset are inaccessible, they manually define a mapping from the language or the tree bank-specific fine-grained tagset to the universal tagset [31]. POS tag accuracies for 25 different treebanks was an experiment in their work to evaluate POS tagging accuracy on a single tagset. And they combined the cross-lingual projection POS taggers [59] with grammar induction system [60] which needs a universal tagset to give an unsupervised grammar induction system for multiple languages.

Zeman and Resnik worked on Interset Project which used in cross-language parser adaptation [61]. In this approach, a tagset of a language is converted into the universal tagset using encoding algorithm implemented in the support library. The above project serves as an intermediate step on the way from tagset A to tagset B. They have covered 20 tagsets in 10 languages. Zeman and Resnik [61] claimed that their approach differs from Google universal tagset approach as McDonald et al [31] did not want to learn the details of existing tagsets more deeply because they eliminate most of the language-specific information, except for the core parts of speech that they find universally. In contrary, Interset eliminates as little as possible because they kept what they find anywhere. Direct conversion from one language to another language didn't focus on this approach.

An international collaborative project called "Universal Dependencies project" proposes a scheme for the treebank annotation, which is suitable for a wide variety of

languages and assists cross-linguistic study [62]. The universal annotation guidelines which are built on Google Universal Part of Speech tagset for POS, the Interset framework for morphosyntactic features and Stanford Dependencies were created by them ([63]- [64]) for dependency relations [62]. Forty languages are covered in the current version 1.3. But in this approach also, they did not focus on the direct conversion from one language to another language.

Majority of researchers have focused on mapping several tagsets to a universal tagset using the guidelines developed. Despite the standards, researchers kept introducing tagsets which posed key challenges for standardization using universal tagset. As POS tagsets become widely used, there is a growing need for aligning tagset between multiple languages and need of aligning multiple tagsets to one tagset [65].

But it is a specific aspect as researchers kept developing new POS tagsets by considering morphosyntactic features deeply despite the standardization of POS tagsets. By adaptation of knowledge from the ontology alignment and schema alignment, this paper focused on the tagset alignment among languages. Earliest schema integration merged a set of given schemas into a single global schema [66] As databases became widely used, there was an emerging need to translate data among multiple databases. As a result, many researchers focused on the alignment between different schemas. In the ontology alignment also, researchers matched entities to determine an alignment between different ontologies. Most of these approaches are semi-supervised as they could not receive the best output by using automatic process. So in this paper also, the focus is based on semi-automatic process.

2.7. Existing machine translation systems using hierarchical phrase-based model

This section reviews the literature about adding hierarchical phrase-based model Statistical Machine Translation system and existing Machine Translation systems for Tamil, English, Malayalam and Sinhala languages. We conducted experiments with hierarchical phrase-based translation using Moses, for the translations between Tamil-English, Malayalam-English and Tamil-Sinhala languages and compared the results with traditional phrase-based models with the same corpora. We have selected Tamil-Sinhala pair of languages to check the hierarchical model, which has the same

sentence structure. The hierarchical model is chosen to overcome the word reordering issue in the translation.

Mahsa Mohaghegh and Abdolhossein Sarrafzadeh [72] adopted this method for the translation between English and Persian languages. As they have observed several challenges when they translated between English and Persian languages, they have moved to hierarchical phrase-based translation. Joshua and Moses toolkits were used in their work. For the English to Persian direction, IRNA monolingual corpus with about 6 million sentences was used. The best result claimed in the paper is 4.5269 NIST and 0.3708 BLEU using the Joshua based system trained on 50K corpus [72]. They have concluded that hierarchical decoder Joshua captured word order better than Moses and they could observe better translation in the English to Persian direction. In this approach, they did not focus on South Asian languages.

One of the early works on hierarchical phrase-based model in SMT for south Asian languages was by Jawaid et.al [73] who examined between English and Urdu. They experimented using the Moses SMT system and presented an Urdu aware approach based on reordering phrases in the syntactic parse tree of the source English sentence [73]. All together they could collect 29,322 parallel sentences to train the system. The monolingual corpus was in Urdu language and collected corpora contain around 61.6 million words in around 2.5 million sentences [73]. A traditional phrase-based translation model with the bidirectional reordering model is used as baseline setup. The results are based on BLEU score. They got highest BLEU score as 25.15 which shows 3.54 score improvement compared to traditional SMT.

Nadheem Khan et al. [74] have focused on English to Urdu HPM SMT. Moses and GIZA++ tools were used in the experiment. EMILLE database with 6596 sentences parallel corpus was used for training. 825 sentences were used for tuning and 824 sentences used for testing the system. Target parallel corpus of Urdu, the corpus of Quran and corpus of Bible used as monolingual corpora with 40,000 segments. The k-fold cross-validation method was used for sampling of the corpus. Here k=5 was selected by taking 4/5 of the total corpus as training and 1/5 as tuning and test set for an experiment on all folds. They have evaluated results using NIST and BLEU scores [74]. Highest BLEU score of the result was 0.29 in the experiment

using hierarchical model [67]. But at that time they got 0.40 as BLEU score using traditional SMT. So they did not get better result to compare to the traditional method.

In this work, we are not only focusing on Tamil-Sinhala language pair. By literature survey, we could understand that most south Asian languages also do not use hierarchical model. So we have applied hierarchical phrase-based model for the translation between Tamil-English, Malayalam-English and Tamil-Sinhala languages. We have already seen the literature survey of Tamil-Sinhala translation system. Below brief literature survey of Tamil-English and Malayalam-English translation approaches are discussed. Various works with different approaches have been proposed for translation between Tamil-English and Malayalam-English. The following fragment will provide the significance of machine translation and will identify a place where a new contribution could be made for those languages by analyzing published information in the area of machine translation.

Ulrich Germann [75] conveyed his experience with building a SMT system for translation between Tamil and English from scratch, including the creation of a small parallel Tamil-English corpus. Following this research, there are several other types of research [76], [49] using traditional SMT. Loganathan developed SMT system by integrating morphological information. He separated the morphological suffixes to improve the quality of traditional phrase-based model [77]. Anandkumar et al. [32] adopted factored SMT system to handle the morphologically fluent Tamil sentences. They applied the manually created reordering rules to the syntactic trees for rearranging the phrases in English. This improves the performance in local distance sentences and already available sentences in the training corpora [78]. But long-distance reordering and new sentence reordering are not handled in these approaches.

First effort 'Rule-Based translation system' reported in the translation from Malayalam to English [49]. But, development of rule-based systems requires more cost, time extensive linguistic rules and it sometimes fails to find good translation due to search errors during the decoding process. Sebastian et al. [79] proposed a SMT approach by adding some pre-processing and post-processing steps. Alignment model is increased by adding the parts of speech information into the bilingual corpus and removing the inappropriate alignments from the sentence pairs. Corpus is pre-processed by suffix and stop word elimination techniques. They have used order

conversion rules to resolve the structural difference between English and Malayalam languages [79]. But, adding rules to translation also faces problems such as high cost in formulating rules and conflicts when the numbers of rules are increasing.

2.8 Existing machine translation systems using factored phrase-based model

As we saw in the above section, except one system [9] all other systems did not focus on adding linguistic information in the translation between Tamil and Sinhala languages. But Pushpananda et al. [9] also did not focus on factored model translation systems. So this section is focusing on various approaches in the factored model used for different language pairs.

The word-based model proposed by Brown et al. [3] is the foundation of all statistical translation models. Then it is extended to phrase based and syntax based techniques. In the beginning, phrase-based translation has seen in the alignment template model that was introduced by Och et al. [71] Marcu et al. [80] proposes joint probability model for phrase-based translation. A significant heuristic approach is proposed by Koehn et al. [71] to extract phrases which are consistent with bidirectional word-alignments generated by the IBM models [71]. This approach shows better performance than syntactically motivated phrases, joint model and IBM model 4. In the factored model, we can add more than one linguistic feature as factors. Most of the researchers focused on POS integration and morphology integration. But this research is focused on POS integration. So this section will mainly focus on POS integration than morphology integration.

First, POS integration approaches are described. Rottmann and Vogel [81] used Parts of Speech information to reorder source side sentences in the SMT. From the word aligned corpora, reordering rules are learned. They have integrated a lattice which contains all word reordering rules in the decoding stage. Different reordering rules have different probabilities. They have added context information also in the reordering rules. Reordered source corpus was used to better capture the reordered word sequence at decoding time. The experiments are based on English → Spanish and German ↔ English translations using European Parliament Plenary Sessions corpus. The results showed that their approach overtakes previous word reordering strategy, which used only distance information.

Kaeshammer et al. [84] presented an extended version of phrase-based SMT models by incorporating Parts of Speech information. Scores are summed to the traditional phrase table which represents how the phrases correspond to their translations on the part-of-speech level. Two different scores learned from a POS-tagged parallel training corpus are used in their approach. German and English language pair was considered in their experiments. Their experiments showed that extended models achieve similar BLEU and NIST scores compared to the standard model. Additional manual investigation reveals local improvements in the translation quality.

Dhanesh [85] analyzed the problems in translation and proposed an automated method for English to Tamil translation to solve the problems. Like any rule-based system, the developed system parses the input sentence, reorders to obtain the target phrasal structure, replaces the words of the sentence with its target equivalence, and lastly synthesizes the target word to get the complete word form.

Ueffing et al. [86] investigated methods to improve quality of translation between morphologically rich language and morphologically poor language. They have used part of speech information and maximum entropy modeling in their method. Experiments were applied on English into Spanish and Catalan on the LC-STAR corpus.

Morphology integration in SMT has been reported in [87], [88], [89], [90], [91], [92], [93], [94] and [33]. Koehn et al. [87] proposed factored translation models by combining feature functions to handle linguistic information in a log-linear model. Nießen et al. [88] showed that the use of morphology information in the corpora drastically reduce the need of parallel training corpora from their experiments. A novel algorithm to combine morphological knowledge was proposed by Panagiotis [89]. The languages they have focused on were English and Greek. The word stems acquired automatically using an unsupervised morphological acquisition algorithm were incorporated with SMT. Linguistica system was used to perform morphological analysis for both source and target languages. Adding linguistically motivated syntactic features to particular phrases and improving morphological agreement in machine translation output by post-processing approaches are introduced by Soha Sultan [90]. The syntactic features she considered were part of speech and dependency

parse tree. The languages used in the research were Arabic and English. Adri`a de Gispert Ramis [91] improved the performance of SMT systems using morphosyntactic information. They gave additional linguistic information beyond the surface level in both target and source side into SMT system. Through an additional verb instance model, he proposed a translation model tackling verb form generation. The experiments were based on English and Spanish languages.

However, there has been any integration of POS and morphology into SMT stated in the literature for the Sinhala-Tamil language pair. Therefore this empirical research is expected to be supportive to come up healthier approaches to build a successful Machine Translation system for translating between two morphologically rich and low resource languages.

When we come to POS integration, as Sinhala and Tamil languages are low resources, we had interoperability issue. To overcome this issue, we came up with semi-automatic alignment algorithm to align different POS tagsets and we studied about language divergence. To align POS tagsets we need to come up with a better POS tagset and tagger.

2.9 Existing Translation Systems Using Chunking the Words

To overcome challenges such as unknown words, context awareness, better word choice, word flow, ambiguity in translation, translating name entities, translating abbreviation and initials and mapping one word with one or more words, we have improved our SMT system by preprocessing based on chunking. In this section, the existing approaches using chunking in SMT are described.

Arora and Agrawal [95] focused on preprocessing techniques which cover punctuation symbols, casing, word spellings and their normalization and handling of numbers and named entities (NEs) for English-Hindi corpus. The best performance is reported by them with retaining the punctuation symbols, lower-cased English corpus and spell normalized.

Yu Zhou et al. [96] proposed a new algorithm “Multi-Layer Filtering” for automatically extracting bilingual alignment chunks from the parallel corpus. They have focused on Chinese-English language pair for their experiment. According to different features of chunks in the parallel corpora, multiple layers are used to extract

chunks in both languages. Those chunks are one-to-one matching to each other. The Multi-layer algorithm doesn't depend on any information of tagging, parsing, the syntax analyzing or segmenting for Chinese corpus [96]. They showed that this algorithm achieves better performance compared to traditional SMT system. The accuracy of chunking in the term of F-score is 0.70. Chunk-based system achieved 0.290 BLEU score while word-based system's BLEU score is 0.259.

Santanu Pal et al. [97] used chunking method to overcome the reordering issue in traditional SMT system. They proposed a method to efficiently handle reordering between distance language pairs in phrased based SMT using chunking. They prior reordered the source text at chunk level to stimulate the target language to overcome the reordering problem. Target word order suggested by word alignment is followed as next step. The test set is reordered using monolingual MT trained on the source and reordered source. They compared this approach with pre-ordering of source words based on word alignments and the traditional approach of prior source reordering based on language-pair specific reordering rules. For the experiments, English-Bengali language pair was used. The results from the experiment showed that word alignment based reordering of the source chunks gives good performance than other reordering approaches. They got better BLEU score 13.17 using this approach [97]. It is a statistically significant improvement over the traditional phrase-based model.

Arianna Bisazza and Marcello Federico [98]used chunking approach as Arabic-English language pair have a large number of syntactic mismatches due to the wrong long- range reordering of the verb in the sentence. They proposed chunk-based reordering techniques to automatically detect and displace clause-initial verbs in the Arabic language. The training data was preprocessed to collect statistics about verb movements. Specific verb reordering lattices derived from this analysis were applied on the test set before decoding. They have performed the experiments on NIST-MT 2009 parallel corpus. This approach shows better performance in the term of BLEU score. The best BLEU score they got is 48.96 [98].

2.10 Existing Translation Systems Using Segmenting the Words

Both Tamil and Sinhala languages are morphologically rich languages. Treating morphologically complex words (MCWs) as nuclear units in translation would not give a desirable result. But Sinhala language does not have morphological analyzer to

handle morphosyntactic features. When translating across Sinhala and Tamil languages, morphological changes cause out-of-vocabulary (OOV) issue between training and test sets leading to reduced BLEU scores in the evaluation. To overcome this problem we are focusing on segmenting the words. This section describes existing systems using segmentation approach.

Xiaolin Wang et al. [99] focused on optimizing Chinese word segmentation for Chinese and English language pair SMT. They claimed that the research method is independent. An approach based on word splitting with reference to the annotated word alignment. They formulated an approach based on word splitting with reference to the annotated WA to overcome the conflicts arrived when using automated segmenters trained on manually annotated corpora. The results of the experiment showed word segmentation reduced the word alignment with the error rate of 6.82% [99]. They got 0.63 BLEU score improvement compared to other related works when they used same Chinese-English OpenMT corpora.

Rohit More et al. [100] attempted to overcome two well-known issues such as the difference in morphological characteristics of the two languages and data scarcity. They have used “word segmentation” and “pivots” to overcome these issues in morphologically rich languages. They have chosen Hindi and Malayalam as their language pair. Both languages are morphologically rich languages but they belong to different family. Triangulation was used as pivoting strategy in combination with morphological preprocessing. When they combined pivot with direct SMT, they have observed a significant amount of improvement in the translation. When they increase the number of pivots they have achieved more performance. From the experimental results, they came up with a conclusion that segmentation is a must. They attained an improvement of 9.4 BLEU score points which is over 58% compared to the traditional SMT system [100].

Peyman Passban et al. [101] proposed two different methods to convey morphological information for SMT models. In the first model, they introduced a new morphological factor which based on subword- aware word embedding to enrich factored SMT systems. They used a subword-level neural language model to capture sequence and word dependencies. They have done experiments on Farsi and German

languages. Experimental results showed significant improvement by using both methods.

2.11 Summary

This chapter presented the literature survey for available Machine Translation systems between Sinhala and Tamil languages, existing approaches in language divergence, Linguistic tools, alignment between different POS tagsets, MT systems using HPM Model, MT systems using factored phrase based model, translation systems using chunking the words and translation systems using segmenting the words. In Sri Lanka, statistical machine translation methods are frequently applied for these language pairs nowadays.

THEORETICAL BACKGROUND

3.1. General

In European countries, machine translation plays a major role for long period. Huge leaps have been occupied by machine translation in the last decade with the beginning of effectual machine learning algorithms and the formation of large annotated corpora for European languages. Nowadays, NLP research in Indian languages also is in a reasonable place. Considering local languages of Sri Lanka (Sinhala -Tamil) very minimal researches have been carried out so far. Tamil and Sinhala languages which gain importance since both of them are acknowledged as official languages of Sri Lanka. Due to the war situation, only a few people can understand both languages. So the translation between two languages gains importance in the current situation of Sri Lanka. However, currently most of the researches are based on rule based techniques due to less annotated corpora. The requirements for evolving NLP applications in Tamil and Sinhala languages are the accessibility of parallel corpora, tagged corpus, lexical tools such as POS tagger, morphological analyzers and computational models. Lack of parallel data and lack of lexical tools in both languages are the major reasons for the slow growth of NLP work in these languages.

3.1.2 Morphological Richness of Sinhala/Tamil Language

The Tamil language is belonging to agglutinative language. One or more affixes can be attached to the lexical root of Tamil words. Most of the affixes are suffixes which can be categorized into derivational suffixes or inflectional suffixes. Derivational suffixes either change the POS or its meaning. Inflectional suffixes make classes such as a person, number, mood, tense, etc. A word can be extended with a large number of suffixes which require more words and sentences when we translate into the English, due to the no absolute limitation on the agglutination.

Tamil belongs to one of the morphologically rich languages. Suffixes are used to perform the plural marker, postpositions, functions of cases and euphonic increment in noun class. Tamil verbs are inflected for tense, person, number, gender, mood, and voice. Computationally, ten thousand inflected word forms can be made for each root word, out of which only a few hundred will exist in a typical corpus [105]. Tamil is consistently head-final language. The sentences in the Tamil language

belong to a Subject-Object-Verb order. However, it allows word order to be changed also. So it belongs to one of the word order free language.

The morphology of the Sinhala can be described on the basis of different parts of speech. In Sinhala, there are four POS, namely, namapada (noun), kriyapada (verb) nipatha (close to prepositions in English, but not the same) and upasarga. Sinhala nouns have five types of inflections, such as gender, number, person, case and article (definite/indefinite). There are three genders, prusha linga (masculine gender), sthri linga (feminine gender) and napunsaka linga (neuter gender). First person uththma purusha, second person maddyama purusha and third person prathama purusha are the three persons. Also there are nine cases like other Indian languages.

3.1.3 Challenges in Tamil/Sinhala Translation

There are many concerns that make a Tamil-Sinhala translation mission to difficult. These relate to the problems of divergence between languages, morphologically richness and low resource languages. Language computing needs exact representation of context. The natural languages are highly uncertain and vague, so achieving such representations are very hard. The various sources of uncertainties in translation are described below.

Ambiguity in Morphemes

Sinhala and Tamil morphemes are ambiguous in the grammatical category and the position it takes in a word construction. Some grammatical category in one language may not be mapped directly to another language. This mostly happens when a number of aspects used in the specialization between languages. For example, the Sinhala language does not have animate/ inanimate categories in verbs but Tamil does have it. It is also possible that a grammatical category in one language does not occur in another language at all. In this case, we won't be able to map the grammatical category at all. Every language has some specific features. Some words in both languages have ambiguity to classify in a particular category.

Ambiguity in grammatical category of morphemes

A morpheme can have more than one grammatical category. For example, the morpheme **அது** (athu), **ஆன** (ana), **து** (thu) can occur as Nominalizing suffix or 3rdPerson neuter suffix. **யார்** (yaar), **எது** (ethu), **எப்போது** (eppothu) words can become below the relative pronoun or question words according to the context.

Word class Ambiguity

A word may be different in the meaning according to its POS or word class. A word may belong to more than one interpretation. For example, the word **படி**"padi" can take noun class or verb class according to the context. So according to the noun or verb translation may differ.

- padi- study (V) or step (N)

  கீழே படி உள்ளது. (Noun)

  தினமும் பாடங்களை படி. (Verb)

- වී ඇත -paddy(Noun) or happened (Verb)

Word sense Ambiguity

Even though a word associated to a specific grammatical category, it may be ambiguous in the sense. For example, the Tamil word **காடு** "kaadu" has 11 senses in noun class and 18 senses in verb class [106]. For example, the following sentence has two different meanings.

- **அவன் பாடல் கேட்டான்**.

  (He heard the song)

  (He asked the song)

Sentence Ambiguity

A sentence can be ambiguous even if the words are not ambiguous. For example, in the following sentence, we can get two interpretations the following sentence has two interpretations.

- "நான் ஒரு அழகான சிறுவனையும் சிறுமியையும் பார்த்தேன்"

  (I saw the pretty boy and girl)

  (I saw the pretty boy and pretty girl)

The words are not ambiguous but the sentences are ambiguous.

Word order

The generated translations were having same word ordering as of the source, in scenarios where reordering which is technically wrong to write without that reordering. In Sinhala, the salutations/titles (Mr. Mrs. Miss.) are added after the name whereas in Tamil it comes before the names. However, the system can translate into the same word order as of the sources, which is incorrect. This requires attention towards reordering model. In some translations, the flow of sentences may be correct in different ordering styles, where translation and reference may differ from each other in their word order yet both are considered correct. This will reduce the score though it is correct.

Out of Vocabulary

Some wordings are left not translated because they didn't exist in the training corpus. They can be categorized based on nature as follows: The not translated words are being abbreviations, initials or names people, places or organization. This requires looking into different approaches to addressing the integration of terminology and transliteration modules.

Following are few examples of such:

Abbreviation: අ.පො.ස. (සා.පෙළ) – G.C.E. (O.L)

Name: එම්. මනෝජා - M. Manoja

The word may be an inflected form of a word that exists in the corpus. Since the system doesn't incorporate any syntactic analyzers within it, different inflected forms of the same word are considered to be different words. Therefore, they are left not translated.

Translation not appropriate to the context

In instances, translated words were the correct translation of the source though they didn't help to express the correct meaning according to the context. The word 'රූපවාහිනී' in the flowing sentence represent 'television' though it has meanthe ing of a 'television channel' also. This gives the glimpse on more emphasis on the language model on making the translation more contexts aware.

Example: **தொலைக்காட்சி** vs **ரூபவாஹினி** in the sentence "ශ්‍රී ලංකාව තුළ ග්‍රාහකයින්ගෙන් මුදල් ලබා ගැනීමේ පදනම මත (Pay TV) රූපවාහිනී සේවාවන් පවත්වාගෙන යාම"

One word can map with one or more word

Single target (one – many) types of translations gets translate word by word which gives completely wrong meaning. In the translation, there are more than one-word combinations which get translated to single wording in a target based on the context, as of the example of 'give up' in English, which may not give sensible translation if broken into individual wordings. Following source 'ක්‍රියාත්මක වේද' means - "whether it works". When it is translated to Tamil, the first two words should be into a single word as '**செயற்படுத்தப்படுகின்றதா**'. But, the middle word "වේද" can be translated into '**வேதங்கள்**' which meant to be "religion", which is wrong in this context though it is a valid translation if only that word alone is considered.

3.2 Language Divergence

Translation is a highly tough task. It targets at conserving semantic and stylistic equivalents of the source text into the target text. Creating deviations based on the context is a difficult task. When the source sentences are recognized in a different manner in the target language, divergence in translation rises. Different linguistic and extra-linguistic constraints play pivotal roles in translation resulting in divergences and other issues. Appropriate identification and understanding of these issues are

significant in both manual and machine translation. Furthermore, for generating good translation in the target language, the resolve of such problems is a pre-requisite. Divergences happen at different levels and harshly affect the quality of a translation. Dorr [24] suggests ways to see into this aspect of translation in small details between any two languages. Based on this we focus on various divergences observed in the translation between Sinhala and Tamil languages. Divergence might disturb the quality of a translation as a language dependent phenomenon. Language divergence is classified into two broad categories:

a. Syntactic Divergence
b. Lexical-semantic Divergence.

3.2.1 Dorr's classification

Dorr (1994) has recognized seven classes of translation divergences. These classes are:

i. Thematic Divergence
ii. Promotional Divergence
iii. Demotional Divergence
iv. Structural Divergence
v. Conflational Divergence
vi. Categorical Divergence
vii. Lexical Divergence.

3.3 POS Alignment

We briefly introduce the Parts of speech tagset alignment problem in this section by adopting knowledge from the ontology alignment and schema alignment. In the ontology alignment also, researchers matched entities to determine an alignment between different ontologies. But, since direct mapping of same labeled tagsets is not possible in all cases of POS tagset alignment, this is more challenging problem compare to ontology alignment. Most of ontology alignment approaches are semi-automatic as they couldn't receive the best output by using automatic process. So in this research also, the focus is based on semi-automatic process.

The POS tagset alignment problem is to find a set of correspondences between two languages' tagsets P_1 and P_2. Because tagsets can be modeled as trees, the

problem is often cast as a matching problem between such trees. A tagset tree, P, is defined as, P=(V, E), where V is the set of labeled vertices representing the tags and E is the set of edges representing the relations, which is a set of ordered 2-subsets of V.

Definition 1 (Alignment, correspondence Maα). Given two tagsets P_1 and P_2, an alignment between P_1 and P_2 is a set of correspondences: ($x_{a,}$, $y_{\alpha,}$, r) xa ∈ P_1 and yα ∈ P_2 being the two matched entities, r being a relationship holding between x_a and y_α, in this correspondence.

$M_{a\alpha}$: { $x_{a,}$, $y_{\alpha,}$, r}

x_a : { x^1_a , x^2_a , ….., x^s_a }

y_α : { y^1_α , y^2_α , ……, y^t_α }

r {=, ⊆ , ⊇, … }

Each assignment variable $M_{a\alpha}$, in M is the confidence between the alignment of two languages, and x_a is the tag from one language and y_α is the tag from another language. Here P_1 language has ‘s’ no of tags and P_2 language has ‘t’ no of tags. There are many possible relationships holding between x_a and y_α, but they mostly fall into equal and subsumption relationships.

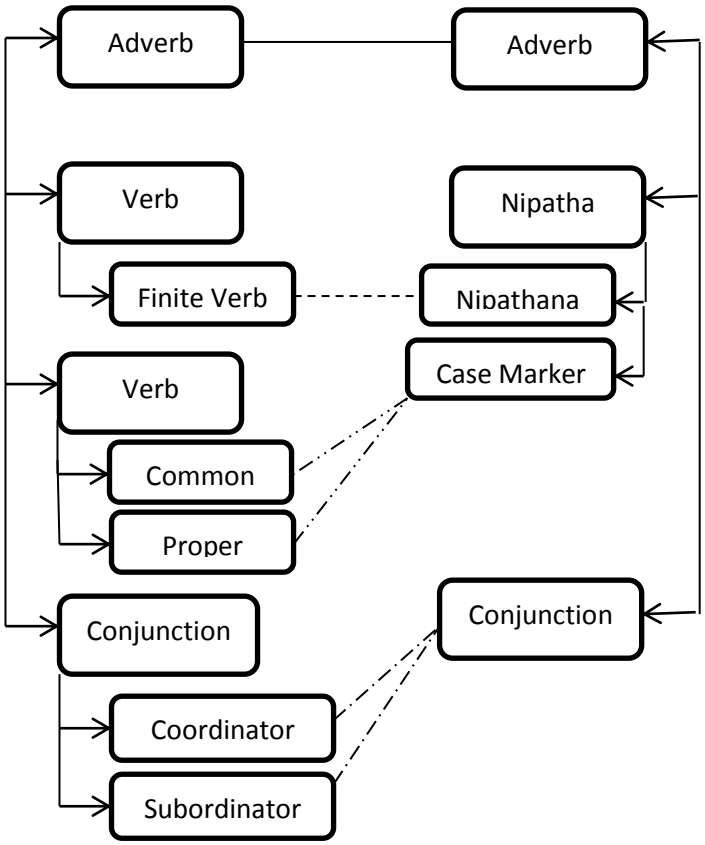


Figure 3.4 Snippet of the alignment between Tamil and Sinhala languages

Equal relationship means one language tagset can equally align with another language tagset. Sometimes a POS tag in one language may not be mapped directly to another language POS tag. This mostly occurs when a number of aspects used in the specialization of a POS tag differ between languages. For example, the Sinhala language does not have animate/ inanimate categories in verbs but Tamil does have it. It is also possible that a POS tag in one language does not occur in another language at all. In this case, we won't be able to map the POS tag at all. Every language has some specific features. But we need to map these kinds of tags as well. If we are not able to find an exact match for a tag, abstract level tagsets can be aligned through the adaptation knowledge of EAGLES guidelines. Figure 3.1 shows the snippet of the alignment between Tamil and Sinhala languages using this semi-automatic algorithm.

3.4 Statistical Machine Translation

As creating and inserting all linguistic rules into a computer would be very tough, statistical approach becomes as favored approach in machine translation field over the

last two decades. SMT is much more dependent on data-driven methods and statistical techniques, aided with the availability of computing power. Translation rules in SMT systems do not have the human intuition by considering linguistic knowledge. They are noisy but they can be created very quickly without spending months or years. It only requires learning parallel corpora to generate translation system. The SMT approach is largely language-independent, that means we can apply SMT to any language pair which has parallel data.

SMT could be expanded by plug and play with new models for preprocessing, post-processing, reordering, and decoding. Defining general "transfer-rules" is a difficult job, especially for languages which share different structures [5]. More computing resources in terms of hardware and software are required for a SMT system. Billions of calculations and probability assignment take place during the training time of SMT system and computing knowledge assists for its high performance. Rule-Based Machine Translation (RBMT) system requires a longer deployment and compilation time. Human-generated rules need to be converted into machine-readable format, so building costs are also greater. But SMT learns statistical patterns automatically from the parallel corpora. As respects to the rules governing the transfer of RBMT systems, surely they can be seen as exceptional scenarios of statistical methods. Yet, they generalize too much and cannot handle exceptions. Syntactic and semantic information can upgrade in the SMT system like the RBMT. A SMT system can improve the translation by retraining or adopting again. In contrast, very similar translation can be obtained after retraining also in RBMT system [5]. Another benefit of Statistical Machine Translation system is that it produces a more natural or closer to the literal target sentence of the source sentence.

The coverage of grammar is also one of the major problems in RBMT. SMT system is a good applicant that meets these criteria. As long as enough the training data is provided to the SMT system, it can learn to have a good coverage. It can statistically model the noise in spoken language, so it does not have to make a binary keep/abandon decision and is, therefore, more robust to noisy data [5].

The initial approach in SMT starts with Brown et al. which was based on the word-based model. Most of these word-based models have been outdated by recent more complex models but they last in word alignment area (Al-Onaizan et al., 1999).

Phrase-based models proposed by Zens et al. (2002); Koehn (2004); Koehn et al. (2007) focused on translating sequences of words in source sentence to target sentence to ensure better translation. Here phrase means a sequence of words, rather than any syntactic phrasal category. Chiang came up with the hierarchical phrase-based model by extending Phrase-based model from sequence of words to a sequence of words and sub-phrases. The hierarchical model brings sub-phrases into existence in order to remove the problems associated with phrase-based MT. It combines the strength of a rule-based and a phrase-based machine translation system.

In SMT system, source language phrases are mapped into target language phrases using statistical methods. Through statistical methods, parameters for the translation are estimated from parallel and monolingual corpora. There are two models such as Translation model and Language model in the SMT system. Translation model is created using parallel sentences and it finds the translation probability between the source and target language phrases. Language model uses the monolingual corpora and it used to ensure the fluent output. It gets the probability of each word according to the n-grams.

There are some translation models existing in SMT system. Some important models are word-based model, phrase-based model, syntax-based model and factored model. Phrase-based model is better than word-based model because 1.Words are not the best atomic unit for the translation (due to frequent one to many mappings) 2.Translating the word groups instead of the single words helps to resolve the translation ambiguities. But small text pieces can be mapped only in phrase-based SMT.

The factored translation is extended version of phrase-based model (Koehn et al., 2003) by representing linguistic information. A word is redefined from a single symbol to a vector of factors. The surface string is a factor for each word, but additional factors can be included as required, in source and target sides. All source factors are specified as input. The target surface factors are the output of the model, while the other target factors are latent variables. Translation is modeled as a process which jointly translates all target factors, conditioned on all source factors. However, surface string output only was concerned; so, by marginalizing the other target factors, we can come up with the conditional probability of target surface string.

3.4.1 Formalism of Statistical Machine translation

Let us now define the phrase-based statistical machine translation model mathematically. First, we apply the Bayes rule to invert the translation direction and integrate a language model p_{LM}. Hence, the best target translation e_{best} for a source input sentence f is defined as

$$e_{best} = argmax_e\, p(e|f)$$

$$= argmax_e p(f|e) P_{LM}(e) \quad (3.1)$$

For the phrase-based model, we decompose p(f|e) further into

$$p(f_1^I|e_1^{-I}) = \prod_{i=1}^{I} \emptyset(\overline{f_i}|\overline{e_i})\, d(start_i - end_{i-1} - 1) \quad (3.2)$$

The source sentence f is broken up into I phrases $\overline{f_i}$. Note that this process of segmentation is not modeled explicitly. This means that any segmentation is equally likely.

Each source phrase $\overline{f_i}$ is translated into a target phrase $\overline{e_i}$. Since we mathematically inverted the translation direction in the noisy channel, the phrase translation probability $\emptyset(\overline{f_i}|\overline{e_i})$ is modeled as a translation from target to source.

Reordering is handled by a distance-based reordering model. Distance-based reordering model considers reordering relative to the previous phrase. We define start i as the position of the first word of the source input phrase that translates to the i^{th} target phrase and end_i as the position of the last word of that source phrase. Reordering distance is computed as $start_i - end_i - 1 - 1$.

3.4.2 Architecture of Statistical Machine Translation

There are three main components in SMT system such as,

1) Translation Model
2) Language Model
3) Decoder

The models are described in below sub sections.

Translation Model

Constructing words with same original meaning and ordering the words in a proper sequence is the capability for translation model. Finding P(t|e) the probability of the target sentence t for given the source sentence e is the important role of the translation model. Sentence aligned parallel corpus is used to train the translation model.

P(t|e) is calculated by counting the number of sentences in t and e in the parallel corpus. But, the challenge is data sparsity. So as the solution, sentence probability is found using the translation probability of the words. The word translation probability is calculated by counting word matching in the parallel corpus. But, parallel corpus is aligned in sentence level; it does not give word level alignments.

If there is word alignment in the parallel corpus, we can exactly count how each word in sentence t is match with sentence e. But here the challenge is how to find the word alignment probabilities in sentence aligned parallel corpus. Expectation-Maximization algorithm is the solution for this challenge. Figure 3.2 shows the alignment of phrases from one language to another language.

Current SMT is based on the insight that a better way to compute these probabilities is by considering the behavior of phrases. In phrase based SMT system, probabilities are calculated by considering phrases matching i.e., single or sequence of words are considered as fundamental units of the translation. In phrase based translation model [71], the goal is to decrease the limits of word based translation by translating unequal sequences of words. The sequences are not technically linguistic phrases. They found using statistical methods from the parallel corpus.

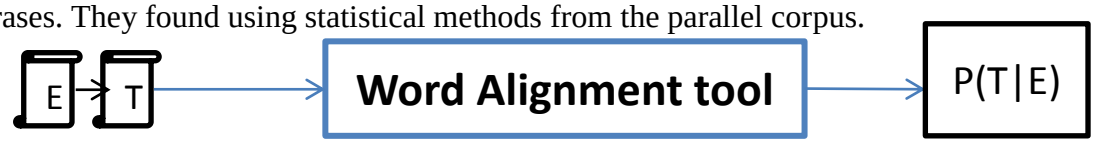


Figure 3.5 Alignment of phrases of both languages E and T

E.g.

මගේ නම ගීතා වේ. **எனது பெயர் கீதா.**

මගේ පොත. **என்னுடைய புத்தகம்.**

Table 2 Snippet of a phrase translation table

Sinhala	**Tamil**	**P(T|E)**
මගේ	எனது	0.66
මගේ	என்னுடைய	0.22
මගේ පොත	எனது புத்தகம்	0.72
මගේ නම ගීතා	எனது பெயர் கீதா	0.22

If the target language ad source language shares the same word order, Phrase-based models work in a fruitful manner. The difference in the order of words in phrase-based models is handled by calculating distortion probabilities. Through the distortion probability, the words are reordered.

Language Model

The fluency of the translated target language sentences is ensured by the language model. Among all possible translations given from translation model, it picks the most fluent sentence with the high value of P(t). The language model can be defined as the model which estimates and assigns a probability P(t) to the sentence, t. Most fluent sentence will get high value for P(t) and least fluent sentence will get low value for P(t). Language model is trained by the monolingual corpus of the target language. It gets the probability of each word according to the n-grams. Standardly it is calculated with a trigram language model.

Example, consider the following Tamil sentences,

ராம் பந்தை அடித்தான்

ராம் பந்தை வீசினான்

Even the second translation looks awkward to read, the probability assigned to the translation model to each sentence may be same, as translation model mainly

concerns with producing the best output words for each word in the source sentence. But when the fluency and accuracy of the translation come into the picture, only the first translation of the given sentence is correct. This problem can be very well handled by the language models. This is because the probability assigned by the language model for the first sentence will be greater when compared with the other sentences. Table 3.2 shows the snippet of the language model.

Table 3.3 Snippet of the Language model

w3	**w1w2**	**Score**
அடித்தான்	ராம் பந்தை	-1.855783
வீசினான்	ராம் பந்தை	-0.4191293

The score is calculated by below equation.

$$P(t) = \prod_{i=1}^{n} P(w_i) \quad (3.3)$$

Where,

$$P(w_i) = \frac{count(w_i)}{count(w_n)} \quad (3.4)$$

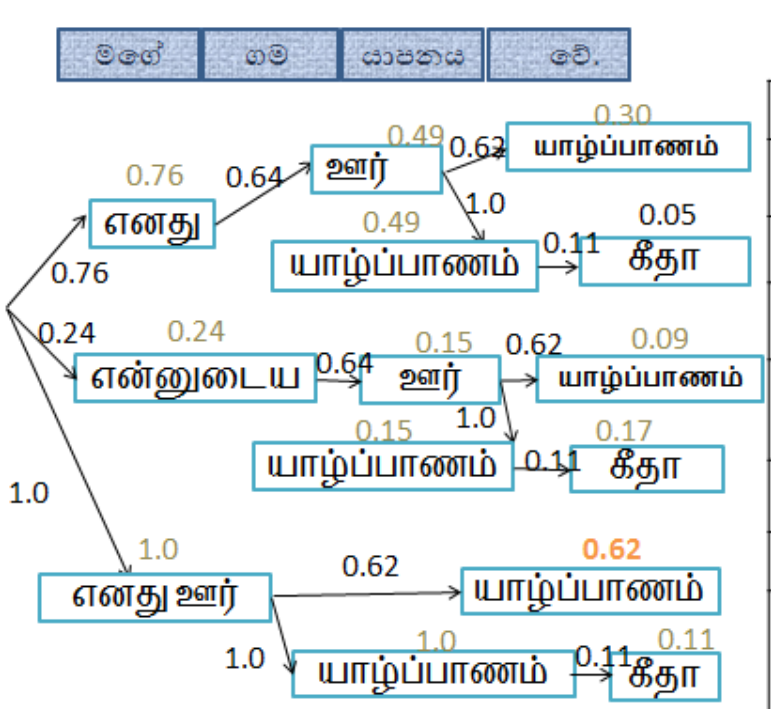
$$P(\text{அடித்தான்} \mid \text{ராம் பந்தை}) = \frac{\text{Count(ராம் பந்தை அடித்தான்)}}{\text{Count(ராம் பந்தை)}}$$

Here I have explained language model for 1-gram. But in this research I have used 3-gram model for the experiments I have done.

The Statistical Machine Translation Decoder

Fining the translated target sentence for the source sentence by using Language model and translation model is the role of the statistical machine translation decoder. Usually, decoding is a search problem that maximizes the translation and language model probability. Statistical machine translation decoders use best-first search based on heuristics. In other words, the decoder is responsible for the search of best translation in the space of possible translations. Given a translation model and a language model, the decoder builds the possible translations and look for the most probable one. Beam search decoders use a heuristic search algorithm that explores a graph by expanding the most promising node in a limited set.

In the figure 3.3 decoding process of statistical machine translation is explained using Sinhala to Tamil translation. A Sinhala input sentence "මගේ ගම යාපනය වේ" is given to decoder. Decoder looks the probability of translation for words/phrases in the phrase table which is already built in the training process. According to the probabilities, it will create tree for all possible translations. In each step probability is multiplied. The highest probability path will be selected as best translation. In this case 0.62 is best path's probability and it will be selected as best translation.



Sinhala	Tamil	p(t|s)
මගේ	எனது	0.76
මගේ	என்னுடைய	0.24
ගම	ஊர்	0.64
මගේ ගම	எனது ஊர்	1.00
යාපනය	யாழ்ப்பாணம்	1.00
වේ	கீதா	0.11
යාපනය වේ	யாழ்ப்பாணம்	0.62

Figure 3.6 Decoding process of Statistical Machine Translation in terms of Sinhala to Tamil translation

3.4.3 Common challenges of SMT system

Parallel corpus and Monolingual corpus are the fundamental sources of SMT systems. Therefore, the vocabulary of the system is closed. Due to this, SMT systems face following a set of challenges.

- Out of Vocabulary: Some words in the source sentences are left as "not translated words" by the MT system since it is unknown to the translation model. The OOV can be categorized as named entities and inflection forms of verbs and nouns.

- Reordering: Different languages have different word ordering (some languages have subject-object-verb while other have subject-verb-object). When translating, extra effort is needed to make sure that the output flow is fluent.
- Word flow: Even though some languages accept free ordering when formulating sentences, according to the order of words, the meaning of sentences may differ. So, we have to be careful when translating from one language to another.
- Unknown target word/words combination to the language model: When the word or sequence of words is unknown to the language model, the system suffers from constructing fluent output as it does not have sufficient statistic on selecting among the word choices.
- The mismatch between the domain of the training data and the domain of interest: Writing style and the word usage has a radical difference from domain to domain. For example, the writing of official letters differs much from that of story writing. And the meaning of words may vary depending on the context or domain. For example, the word 'cell' is translated to a 'small part of the body' if the considered domain is medical while to 'telephone' if the domain is computing
- A multiword expression such as collocations and idioms: Translation of such multi-word expression is beyond the level of words. Therefore, in most cases, they are incorrectly translated.
- Mismatches in the degree of inflection in of source and target languages: Each language has its own level of inflection and different morphological rules. Therefore, most of the time there will not be a one-to-one mapping between these inflections. This creates ambiguity while mapping inflection forms.

Within the above challenges, there are some challenges that are prevalent in Sinhala to Tamil translation. Those are listed below.

- Mismatch of inflection - As Sinhala and Tamil languages belong to different families, there are some conflicts between the inflection forms. Therefore one-to one mapping is not possible all the time.
- Word reordering - Even though, Sinhala and Tamil languages share same sentence structure (SOV), there are some word order different in local context.

- Word flow - Even though Sinhala and Tamil languages accept free ordering when formulating sentences, according to the order of words, the meaning of sentences may differ. So, we have to be careful when translating from one language to another.
- Mapping one word with one or more words – There are some scenarios where two/three words in Sinhala/Tamil language need to be mapped with one word in another language.
- Ambiguity – A word can be translated into different forms according to the context. So, there is ambiguity to select the correct word according to the context.
- Abbreviations and initials – Tamil and Sinhala languages are low resourced. So, having all abbreviations and initials in the corpus is not possible. Some letters should be translating into one letter when we translate from Sinhala to Tamil. So, when we translate from Tamil to Sinhala, there is an issue to select the correct letter.

  E.g ‘Ba’, ‘pa’

  ‘ga’, ‘ha’, ‘ka’

  ‘cha’, ‘sa’, ‘sha’
- Out of Vocabulary – As Tamil and Sinhala languages are low resourced and morphologically rich, it is not possible to have each and every word in the training corpus. So, some words are not translated while we try.
- Low resourced languages – As both languages are low resourced, it lacks in parallel corpora, monolingual corpora and linguistic tools. So, learning all patterns in the translation, capturing all the words and improving via linguistic tools are challenges.
- Orthographical error - As the languages consist of more alphabets than the keyboard system, typing in those languages is a bit complex. In practical use, most of the time non-Unicode fonts are used in document processing, some time with local customization over the font. Though from the point of human reading usage, this makes no harm; this non-standardization in document processing makes it hard to produce linguistics resources for computer processing. In most cases, this conversion process creates orthographical errors in the data.

3.4 Hierarchical Phrase Based Translation

As we see in the previous subsection, phrase based machine translation is based on phrases. To eliminate the challenges in the phrase based translation system, hierarchical phrase model takes sub phrases for the translation. Here, let us see the example of English to Tamil. In the Figure 3.4, we condense this observation into a grammatical rule. A possible grammar rule is that the phrases on whichever side of the word of will be swapped when translating to Tamil. This is the benefit of using sub-phrases. In phrase based translation system, the rotation is fixed only for the particular phrase in the parallel corpus and different rules are needed for different sentences even they follow same structure. So the numbers of redundant rules are increased. In phrase based MT, these redundant rules are stored in a dictionary. On the opposing, hierarchical machine translation replaces these rules by a single rule i.e. every rule is associated with a weight w that values how probable the rule is in comparison to other rules with same rule in the Tamil side.

Hierarchical phrase based translation system combines the strength of rule-based and phrase based translation system. This can be observed from the working of grammar extraction or decoding because hierarchical model uses rules to express longer phrases and phrases as it is for smaller phrases.

For e.g.:- **கல்வித் திணைக்களம்**{kalvith thinaikalam} {Department of Education}➔Department of Education

This example will have a similar expression on the Tamil side but different on the English side. Here X_1 and X_2 are first and second words in the Tamil sentence.

$$X \longrightarrow X_1 X_2, X_2 of X_1 \tag{3.5}$$

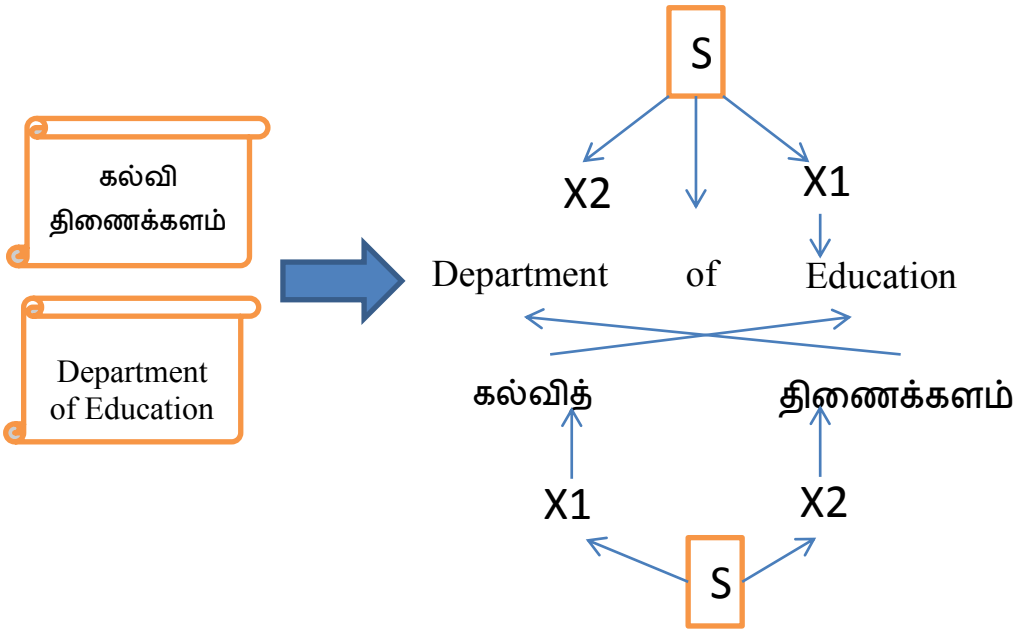


Figure 3.7 Tamil to English translation showing reordering. S=Sentence

Same rules in the grammar are required for parsing and translation. This makes the grammar more interesting. This kind of grammar is formally called synchronous context-free grammar. Synchronization is required between sub-phrases because these sub-phrases need to have a number attached to them since they are essentially all X. X is the only symbol used as a non-terminal apart from the start state S. The numbering system is the way non-terminals are differentiated.

Parser for one language is needed for this approach because all phrases re labeled as X. This is very essential with respect to low-resourced languages since most of the low-resourced languages don't have a well automated parser at the moment. Same distortion model is used by Hierarchical model to reorder the sentences.

3.5 Factored Model

Factored translation models [87] is an extended version of phrase-based model. Vector of factors (word, lemma, Parts of Speech, morphology, etc) substitutes the word. Translation process is done by combining translation and generation steps. Figure 3.5 shows the Redefining a word from a single symbol to a vector of factors.

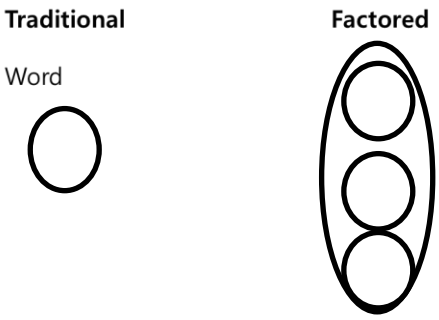


Figure 3.8 Redefining a word from a single symbol to a vector of factors

Additional linguistic features are integrated in the Factored translation model. Those additional word level information is called as a factor. To reorder or grammatical coherence decision, parts of speech information may be helpful. Sparse data problems in morphologically rich languages can be assisted by the translating the lemma and morphological factors separately.

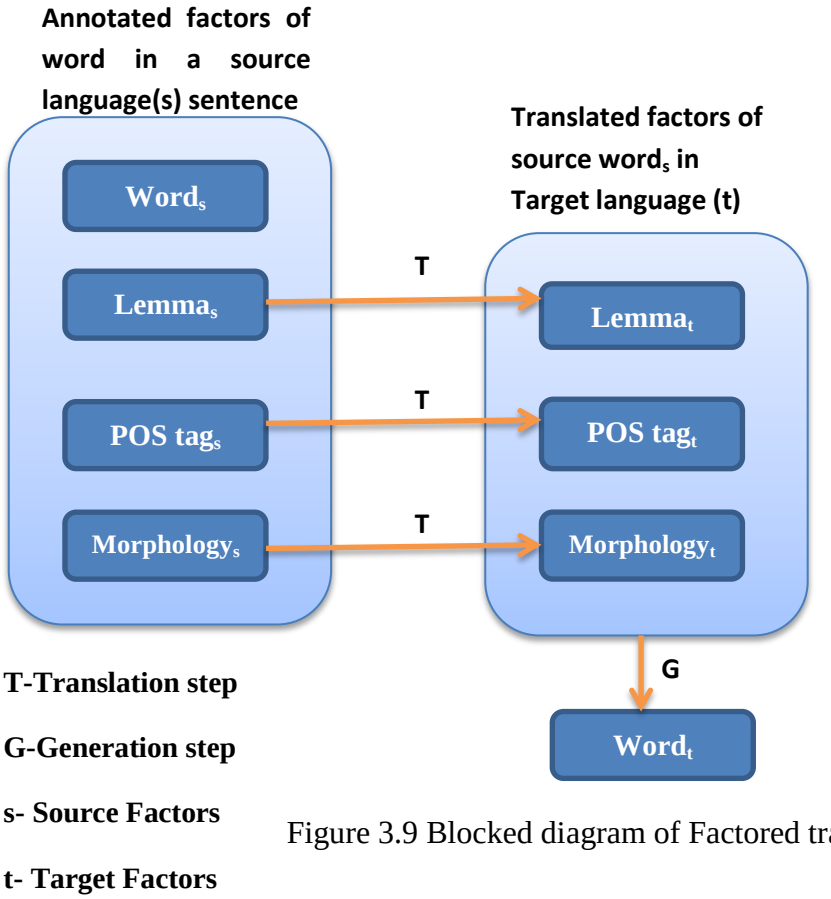


Figure 3.9 Blocked diagram of Factored translation

Translation and generation steps of factored model are illustrated in Figure 3.6. Using factors (lemma, parts of speech, morphology) the parallel corpus is annotated before training the translation model. And additional monolingual corpus can be annotated to train the language model.

The words *ball* and *balls* are completely independent in statistical machine translation. So if there is a word *ball* available in the training corpus, which does not add any knowledge to translate the word *balls*. That means the *ball* can be translated and *balls* cannot be translated by the system. But, this kind of issue does not show up in the English translation systems as English is morphologically poor language.

Hence, if we try to model a translation system between morphologically rich languages on the level of lemmas, different word forms can be derived from a common lemma. In this scenario, we should translate lemma and morphological information separately. Then that information can be combined in the output side to finally generate the output surface word. Such a model can be well-defined straight-forward as a factored translation model.

For factored translation model, annotated parallel corpus is needed. So we need to annotate the corpus with additional factors. For example, if we want to add POS information on the source and target side, we need to have parallel POS tagged training data. Typically this involves running automatic tools on the corpus since manually annotated corpora are rare and expensive to produce. Next, we establish a word alignment for all the sentences in the parallel training corpus using standard methods.

3.6 Chunking

Our procedures can be summarized as follows: First, the most frequent monolingual chunks are filtered from the Sinhala-Tamil parallel texts. This technique allows us to obtain more accurate monolingual chunks and at the same time helpfully makes long sentences shorter; second, sequences of fragments which remain after filtering are simply combined into chunks which can participate in the alignment process. One-word fragments remaining in sentences are treated likewise; finally, in order to guarantee that one Tamil chunk will correspond with one Sinhala chunk, only the best Tamil chunks (those with the highest co-occurrences with Sinhala chunks) are retained for use. This step seems justifiable since translation output quality will not be seriously affected because most of these chunks aligned to one having the same or similar meaning. In the filtering steps, information concerning frequency, n-gram statistics, and mutual information is employed in order to extract bilingual chunks.

PMI based Chunking

On the assumption that the most co-occurring word lists may be a potential chunk, so these word lists are first filtered as initial monolingual chunks. The first filtering step proceeds as follows:

1) As shown in the formula below, D, denotes the degree of cohesion of a chunk which length is k. To some extent, the PMI score of a word lists reflects the probability of that word lists, so this measure is used to define if a word list is a reasonable chunk. D(w_1 ,w_2) (Here k = 2) is first used to compute the PMI score between two adjacent words in sentences, where,

$$D_k = D(w_1 w_2, \ldots\ldots w_k) = (1-\beta) \times MI(w_1 w_2, \ldots\ldots w_k) + \beta \times P(w_1 w_2, \ldots\ldots w_k) \quad (3.9)$$

$$MI(w_1 w_2, \ldots\ldots w_k) = P(w_1 w_2, \ldots\ldots w_k).log \frac{P(w_1 w_2, \ldots\ldots w_k)}{P(w_1)P(w_2)\ldots\ldots P(w_k)} \quad (3.10)$$

Here $MI(w_1 w_2, \ldots\ldots w_k)$ denotes the mutual information of the sequential words $(w_1 w_2, \ldots\ldots w_k)$, $P(w_1 w_2, \ldots\ldots w_k)$ denotes the probability of the sequential words$(w_1 w_2, \ldots\ldots w_k)$; and β is a coefficient between 0 and 1.

2) After computing all the cohesion degrees between any two adjacent words in all sentences, tag the lowest n values as anchor points within the sentences. Scan the sentences from the anchor points forward and backward in steps from 2 to 1 and keep the most frequent initial chunks in each step (where the maximum length of both Tamil and Sinhala chunks are 2, and n is determined by formula
3) The maximum length is defined as 2 because even in a very large training corpus chunks with length 3 are too infrequent. Moreover, the chunks to be obtained should conform to the following principles:

Table 3.3, an example is given to explain the first filtering process in detail. Here, last column values denote the PMI score between two adjacent words

Table 4.PMI score between two adjacent words

Word 1	Word 2	PMI score
ஆளுனராக	இருந்தவரின்	15.734968884109453

உத்தியோகத்தர்களும்	தமக்குரிய	15.734968884109453
உற்பத்தியாளர்களின்	உற்பத்திகளின்	15.734968884109453
உற்பத்தியாளர்கள்	அநேகமானோருக்கு	15.734968884109453

3.7 Segmentation

Sinhala and Tamil languages are morphologically rich languages. Translation between two morphologically rich languages is still uncommon. But translating from morphologically rich language (e.g. Tamil) to morphologically poor language (e.g. English) and vice versa is widely studied the problem in the literature. According to literature, there are some approaches to translate between morphologically rich languages. Morphological analyzer and parts of speech tagger are used to integrate the morphological information into machine translation in most of the researches.

In this approach, we have used Morfessor algorithm to segment words into morphemes in both Sinhala and Tamil languages. Morfessor is a group of methods for unsupervised morphological segmentation. Models of the Morfessor group are generative probabilistic models that predict compounds and their analyses (segmentations) given the model parameters. The cost function of Morfessor Baseline is derived using maximum a posteriori estimation. That is, the goal is to find the most likely parameters θ given the observed training data D_W:

$$\theta_{MAP} = argmax_{\theta}\, p(\theta)p(D_w|\theta) \quad (3.11)$$

Thus we are maximizing the product of the model prior p(θ) and the data likelihood $p(D_W \mid \theta)$. As usual, the cost function to minimize is set as the minus logarithm of the product:

$$L(\theta, D_w) = -log_p(\theta) - log_p(D_w|\theta) \quad (3.12)$$

The data likelihood is calculated using a hidden variable which consists of the currently selected analyses in training time. Next, it is expected that the constructions in a compound occur independently. This simplifies the data likelihood to the product of all construction probabilities in the chosen analyses. Unlike previous versions, Morfessor 2.0 includes also the probabilities of the compound boundaries in the data likelihood. A diagram of the morfessor tools is shown in Figure 3.7.

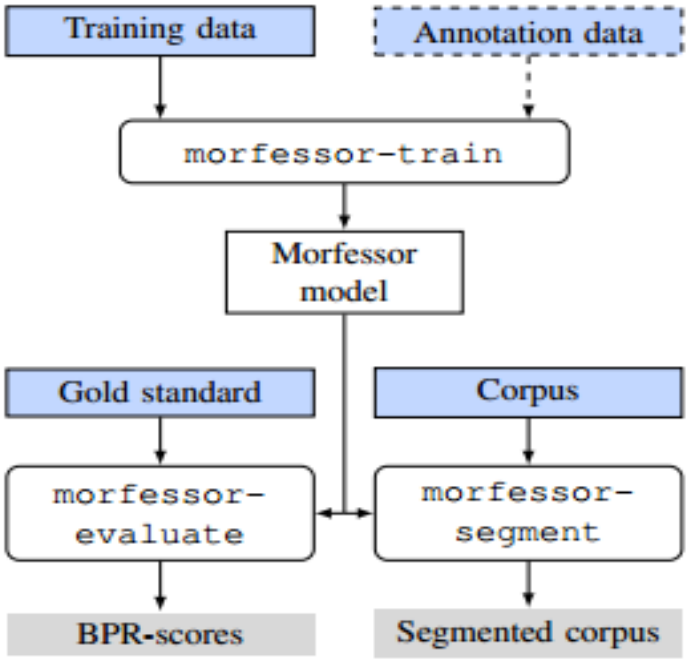


Figure 3.10 The standard workflow for Morfessor command line tools

3.8 Evaluating Statistical Machine Translation

This section provides evaluation methods to find the quality of machine translation system. Evaluation of MT is a very active field of research. There are two main types of evaluation techniques in MT which are automatic evaluation and manual evaluation. This shows how to evaluate the quality of MT system by automatically and manually. The most consistent method for evaluating adequacy and fluency is through human evaluation. But human evaluation process is expensive and time-consuming. The judgments of more than one human evaluator are usually averaged. A quick, cheap and consistent approach is required to judge the MT systems. A precise automated evaluation technique would require linguistic understanding. Methods for automatic evaluation usually find the similarity between the translation output and one or more translation references.

3.8.1 Human Evaluation Techniques

Statistical Machine Translation outputs are very hard to evaluate. To judge the quality of translation one may ask human translators to find the scores for a machine translation output or compare a system output with a gold standard output. This gold

standard outputs are generated by human translators. In the human evaluation, different translators translated the same sentence in different ways. There is no single correct answer for the translation task because a sentence can be translated in different ways. The reason for translation variation is choice of words, word order, and style of translators. So the machine translation quality is very hard to predict.

Table 3.4 shows the scales used for evaluation when the language being translated between Tamil and Sinhala in this research. Using this scale, the judges are asked to assign a score to each of the presented translations. Accuracy and fluency are a widespread means of doing a manual evaluation.

Table 5 Scales of Manual Evaluation

Rating	Description
4	Very Good
3	Good
2	Acceptable
1	Bad

3.8.2 Automatic Evaluation Techniques

The automatic evaluation is the method which uses a computer program to judge the translation output is better or not. Currently, automatic evaluation metrics are widely used to evaluate machine translation system. These systems are an upgrade based on the rise and fall of scores in this automatic evaluation. The major advantage of this technique is time and money. It requires less time to judge a huge amount of outputs. In situations like everyday system evaluation, human evaluation can be too expensive, slow, and inconsistent. Therefore, an automatic evaluation metric that is reliable and very important to the progress of Machine translation field.

BLEU Score

The first and most widely-used first automatic evaluation measure is BLEU (Bi-Lingual Evaluation Understudy) [11]. It was introduced by IBM in Papineni et.al. (2002). It finds the geometric mean of modified n-gram precisions. BLEU considers not only single word matches between the output and the reference sentence but also n-gram matches, up to some maximum n.

$$BLEU = BP.\exp(\sum_{n=1}^{N}\frac{1}{n}logp_n) \quad (3.135)$$

3.9 Si-Ta System

Si-Ta system is developed by the University of Moratuwa for the department of official languages. Si-Ta is a Machine Translation system for Sinhala and Tamil languages which focused on official government documents, with post editing support to correct the translation. Currently, short documents (up to two pages) are targeted by the system. The translation process starts with the input of the source document, either in Sinhala or Tamil. The source document is translated by the Si-Ta system, and some words are highlighted if those words could not translate by the system. Those words and any other translation errors can be manually translated by the human translator. Hence, instead of having to translate a document from scratch, Si-Ta allows human translators to be proofreaders, where they simply have to fix the issues they see in the output.

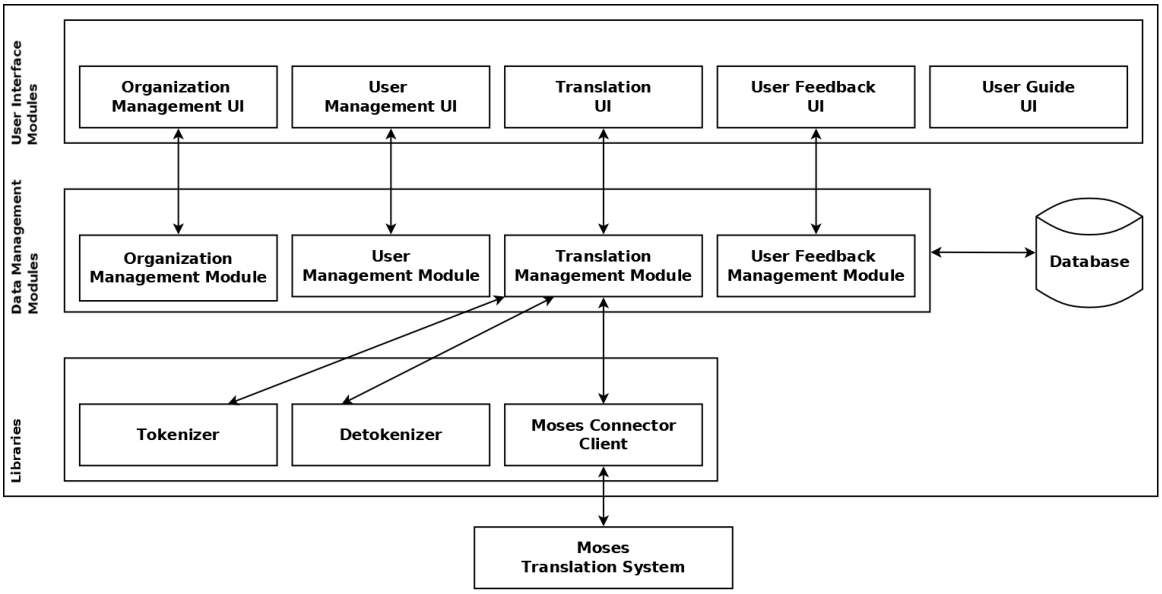


Figure 3.11 Architecture of Si-Ta System

Si-Ta employs simple client-server architecture, with a user-friendly web interface. Different machine translation system can be plugged in the back end without affecting the translator's interface. Si-Ta system is currently used by two different government institutions. The accuracy of the system measured by BLEU score and the current system is reported as 25.05 and 32.85 for Sinhala-Tamil, and Tamil-Sinhala, respectively. Human translators reported an accuracy of 3.32 on a scale of 1-5, which specifies that the translation output conveys the intended meaning, though some amount of post editing is required. This shows the practicality of using Si-Ta system as a Computer Assisted Translation system for Sinhala and Tamil official government documents.

The architecture of the system is showed in Figure 3.8. Simple client-server architecture has been used in the design. To achieve separation of concerns, each of UI modules is connected to a separate module at the data management back-end. Most importantly, the MT system is well separated from the other components, which allows us to experiment with other MT systems without having to modify the rest of the system.

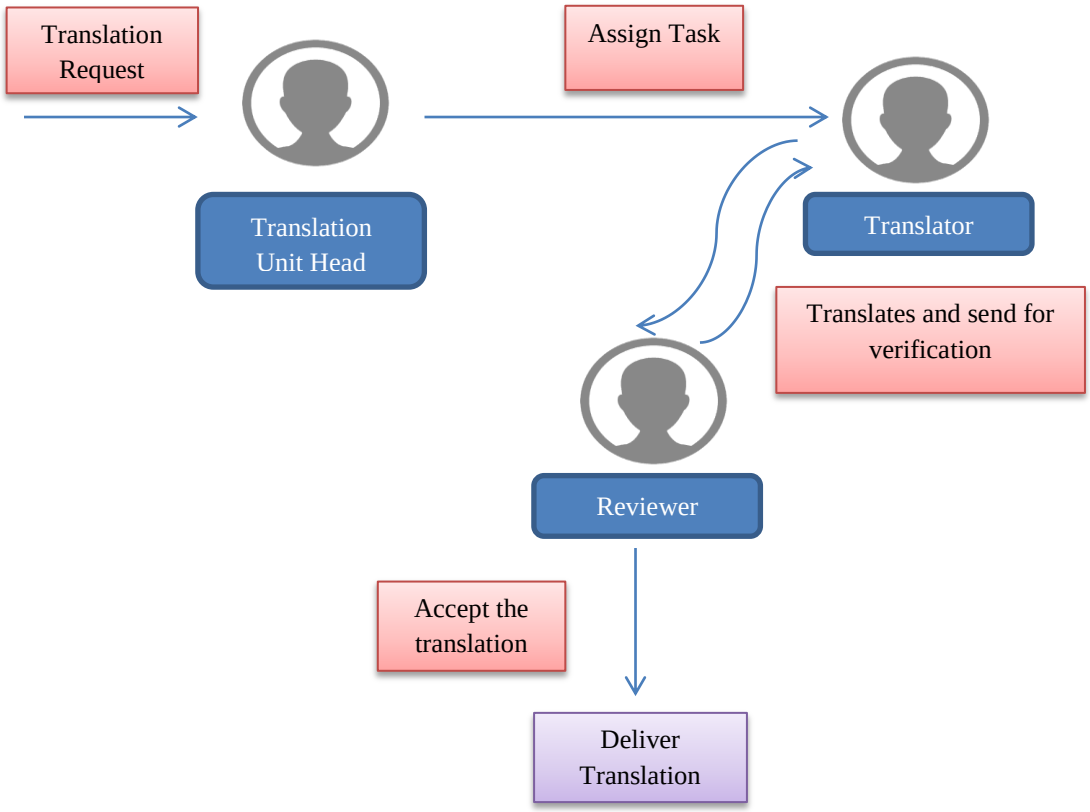


Figure 3.12 Workflow diagram of Si-Ta System

Figure 3.9 shows the workflow diagram of the Si-Ta system. Translation workflow of the current Si-Ta system is catered to the needs of the Department of Official language that uses Si-Ta. However, it is fairly simple to change the same into a different process. The process starts with the head of Translation assigning translation work to the individual translators. The course text can be either typed or imported from a file. The translator enters the source text into the input box, and Si-Ta automatically detects the source language and carries out the translation. Any untranslated words are highlighted, and the translator is able to edit this translated output. The translated document can be sent for the verification after editing. The reviewer can either accept the submitted translation or send it back to the translator with any suggestions to improve the translation. If the translation is accepted by the reviewer, it is sent back to the party that requested the translation (i.e. who sent the source document). Fig. 3.10 shows the user interface (UI) of Si-Ta system when being used for translation.

Totally 23,006 sentences were in the parallel corpus. Moses toolkit with GIZA++ was used to build the translation model and SRILM was used to build the language model. The weights of each model were adjusted at the time of tuning based on their relevance to the tuning set. Featuring weights tuning was done using Minimum Error Rate Training (MERT) on 100 best translations for a set of 1,000

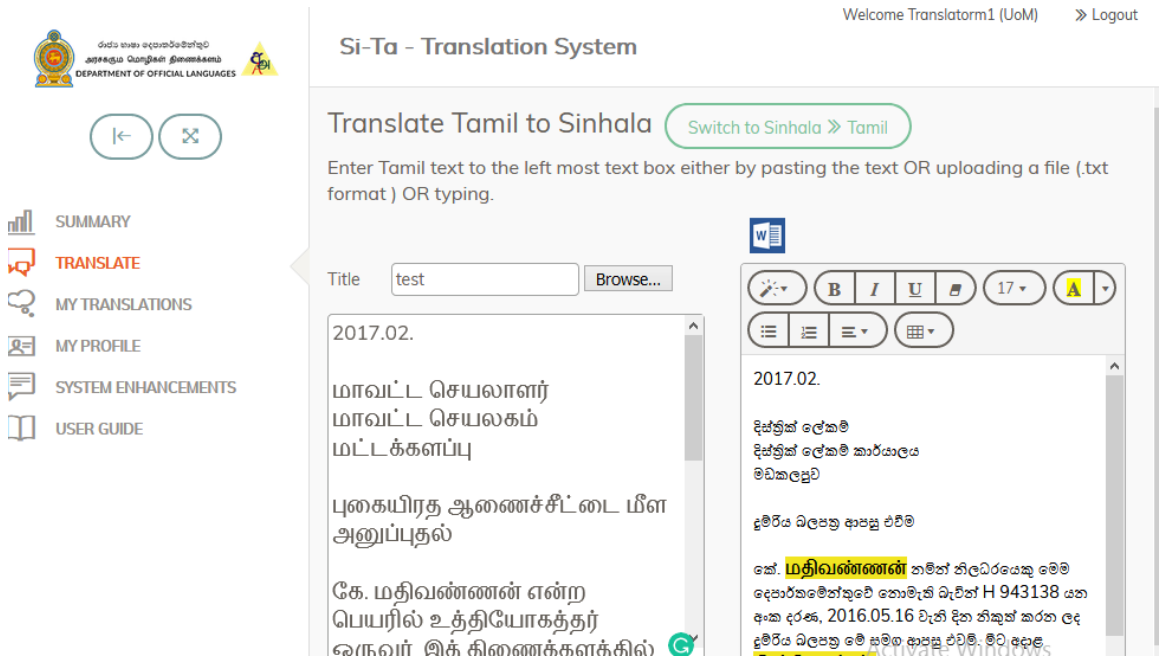


Figure 3.13 User interface of Si-Ta System

randomly selected sentences. 300 sentences are used to test the system. As the contribution to Si-Ta project, I have created Tamil to Sinhala translation model and came up with some solutions as mentioned above to enhance the quality of the translation.

3.10 Summary

Background knowledge of Tamil and Sinhala language processing and approaches of developing linguistic tools are described in this chapter. The morphology richness of Tamil and Sinhala languages also are discussed in this chapter. Background knowledge of the methods to improve the translation is explained. It also describes the background knowledge of the hierarchical model, factored model, chunking, and segmentation to understand the methodology of this research.

METHODOLOGY

The purpose of this chapter is to show the philosophical assumptions underpinning this research, as well as to introduce the strategy of the research and the empirical techniques applied. The scope and limitations of this research also expressed in this chapter. The philosophical assumptions underlying this research come from the interpretive tradition. The methodologies to improve the translation from Sinhala to Tamil are discussed in this research. We have focused on four main methods to improve translation. To come up with the methods to improve the translation, the challenges in the current machine translation system should be identified. We chose Si-Ta, the state of the art machine translation system for Sinhala -Tamil for this study. As a step towards understanding the translation challenges between Sinhala-Tamil, we studied the divergence between these languages. With the analyzed results of language divergence, a divergence among Tamil and Sinhala POS tagsets could be identified. Accordingly, we have come up with an algorithm for the alignment of different POS tagsets.

Based on the analyzed results of language divergence, we could identify translation challenges such as reordering, abbreviations and initials, word flow of the sentence, data sparseness and mapping one word with one or more words. Subsequently, we have used methods such as hierarchical phrase-based model, Factored model with POS integration and preprocessing techniques to address the mentioned challenges. Further pre-processing techniques based on chunking and segmentation. The sections below, we present all the methodologies in detailed manner.

This chapter is divided into six sections. In the first section, the methodology used to identify the divergence between Tamil and Sinhala languages are presented. The next section is about the semi-automatic alignment between different POS tagsets. It describes the methodologies of tagset selection and semi-automatic algorithm for aligning the tagsets. The next section is about the methodologies used to build the hierarchical phrased based machine translation system for low resourced languages. POS integration to SMT system is described in next section. We have integrated the POS information to enhance the quality of the translation. Preprocessing techniques based on chunking to overcome the challenges in the traditional SMT is presented in the fifth section. This section further divided into three sections such as PMI based

preprocessing, NER based preprocessing and POS chunking based preprocessing. Finally, section six deals with the preprocessing techniques based on segmentation. This section describes sub-word building and integrating these sub-words to the SMT system.

4.1 Language divergence between Sinhala and Tamil languages

The methodologies to identify the divergence between Sinhala and Tamil language are discussed in this section. This section provides the systematic method for the identification and probable solution of the lexical-semantic divergences between Sinhala and Tamil.

The classes of translation divergence have been defined according to different types of translation divergences found in a pair of translation languages. In this research, we have largely focused on the translation divergences arising out of grammatical aspects of the translation languages. For identifying divergence in these languages, we have analyzed the results of built Sinhala to Tamil statistical machine translation system. Translation challenges in the system are identified to come up with the divergence between these languages. Then the research is assisted and cross-checked by bilingual experts to finalize the divergence classes. This research is based on Dorr's classification. We have adopted the classes which can be applicable for Sinhala and Tamil languages. The divergences do not belong to any category proposed by Dorr [24], we grouped them separately. Given an input Sinhala sentence and corresponding Tamil sentence, the proposed technique aims at recognizing the occurrence of divergence in the translation. An algorithm has proposed to identify the language divergence according to the definition of those categories.

4.2 Semi-Automatic Alignment of Multilingual Parts of Speech Tagsets

In order to arrive at an agreement between multiple language POS tagset, researchers have adopted various strategies as we discussed in literature review. Some derived a new tagset capturing the morphosyntactic features of some specific set of the languages (Bureau of Indian Standard) and some mapped existing POS tagsets to a universal POS tagset. However, both approaches introduce new POS tagset. Unlike these prior approaches, we took a completely new angle. We cast the problem of heterogeneity in POS tagsets as an alignment of two labeled trees and proposed a novel semi-automatic approach algorithm to solve. We evaluated our algorithm using

a representative POS tagset chosen from Sinhala and Tamil languages. We chose these language pairs since 1. we have access to necessary data and expertise 2. these languages are low resourced 3. they gain more importance as official languages of Sri Lanka. Below the rationales behind choosing the representative tagset from each language are described. Then, semi-automatic POS alignment algorithm is presented.

4.2.1 Tagset Selection

As there are several tagsets available in each language, selection of a POS tagset is essential for this study. While choosing a tagset of a language, the usability and standardization are considered. Following subsections describe the identified POS tagsets of Sinhala and Tamil and how the proper tagset is selected to align.

Sinhala Tagsets

There are two tagsets available for the Sinhala language such as University of Colombo School of Computing (UCSC) tagset developed by University of Colombo [53] and UOM tagset by University of Moratuwa [51]. The details of the tagsets are described in this subsection. UCSC tagset contains 29 tags which include foreign word and Symbol. There are three versions in UCSC tag set.

The University of Moratuwa has built an improved version of UCSC tagset by overcoming the following issues,

1. All Sinhala word classes are not fully covered by UCSC tagset.
2. 3989 words don't fall into any category out of 100,000 words in the manually annotated corpora.
3. Same words are tagged using different tags in different places in the same context.
4. Inflection based grammatical variations don't cover well [51].

There are three levels in this tagset following a hierarchical structure. In sum, they came up with 148 tags. Level I contain the primary top-level parts of speech. Level II tagset is generated by adding inflected forms to Level I. Level II tagset consists of 30 tags [51]. UOM tagset is selected for this study because of the above mentioned major

limitations of the UCSC tagset. Table 4.1 shows the selected UOM tagset at the second level.

Table 6 UOM tagset in two levels

Level I Tags	**Level II Tags**
Nouns	Common Noun
	Proper Noun
	Pronoun
	Noun in Compound Verb
	Questioning Pronoun
	Deterministic Pronoun
	Question-Based Pronoun
Adjectives	Adjective
	Adjectival noun
	Adjective in compound verb
Verb	Verb finite
	Verb participle
	Verbal Noun
	Verb non-finite
	Modal auxiliary
Nipatha	Postposition
	Conjunction
	Particle
	Interjection
	Determiner
	Nipathana
	Case marker
	the preposition in compound verb
Adverbs	Adverbs
	Number
	Abbreviation
	Full stop
	Punctuation

	Foreign word
	Sentence ending

Tamil Tagsets

There are several tagsets available in Tamil language. Selection of a POS tagset is essential for this study. While choosing a tagset of a language, the usability and standardization are considered. This subsection describes the existing POS tagsets of Tamil language. For the Tamil language, there are plenty of tagsets. We considered nine tagsets ([34], [35], [33], [36], [37], [38], [39], [40], [41]) before choosing an appropriate one for this study. Bureau of Indian Standards (BIS) is recommended as a common tagset for POS annotation of Indian languages. Many tags in BIS are same as LDC-IL tagset. It groups unknown, punctuation and residual into one tag. It has 11 tags in level I and 32 tags in Level II tags. Level II is made by further subdividing the level I tags. We choose BIS Tamil Tagset since it is the officially accepted standard tag set for Tamil language. Table 4.2 shows the selected BIS tagset at the second level.

Table 7 BIS tagset in two levels

Level I	**Level II**
Noun	Common noun
	Proper noun
Pronoun	Personal Pronoun
	Reflexive Pronoun
	Relative Pronoun
	Reciprocal Pronoun
	Question words
Demonstrative	Deictic
	Relative
Verb	Verbal participle
	Verb Finite
	Verb Auxiliary
	Infinite Verb
	Conditional Verb
	Relative Particle Verb
	Verbal Gerund
	Verbal Noun
Adjective	Adjective
Adverb	Adverb
Postposition	Postposition
Conjunction	Coordinator
	Subordinator
Particles	Default

	Classifier
	Intensifier
	Interjection
	Negation
Quantifier	General
	Cardinal
	Ordinal
Residuals	Punctuations
	Unknown
	Foreign
	Echo words
	Symbol

In our approach, the third level of both language tagsets is not considered. The third level captures inflection based grammatical variations of the language. We chose to omit Level III for following reasons.

1) It has no apparent impact in most of the applications it used.
2) The deeper levels are at times inflectional forms than being truly POS classes
3) Tagging time increases as we need to split the word into morphemes
4) A large number of tags will lead to more complexity which reduces the tagging accuracy [42]

4.2.2 Semi-automatic algorithm for POS Tagset Alignment

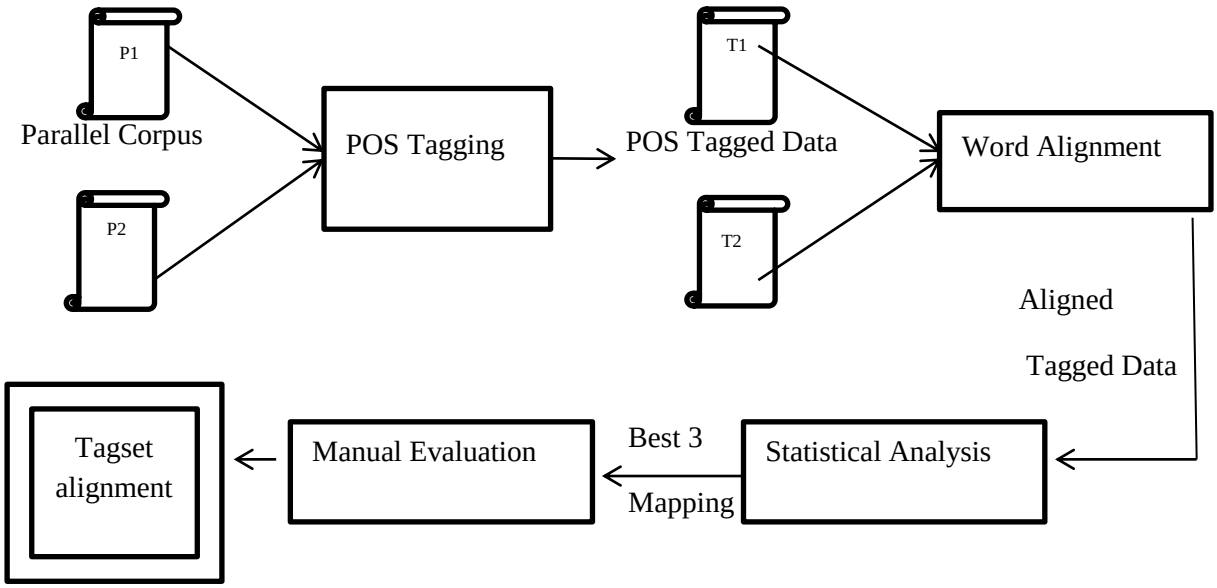


Figure 4.14 Work flow of the semi-automatic POS tagsets alignment of P1 and P2 languages. T1=POS tagged data in P1 language, T2= POS tagged data in P2 language

We proposed a semi-automatic approach for the tagsets alignments. Figure 4.1 describes the workflow of the semi-automatic POS tagsets alignment. The proposed semi-automatic approach requires parallel corpus. Parallel corpus of Languages P1 & P2 was annotated using respective automatic POS taggers. Then the tagged parallel corpus was word aligned using a word alignment tool. Afterwards, best three mappings for each POS tag were selected based on the amount of word alignment and presented to human evaluators. The experts pruned the provided mappings and arrived at a final quality and complete alignment. Below we present the each and every workflow steps and tools used for this approach in a descriptive manner. The experiments are applied on Sinhala and Tamil POS tagsets.

4.3 Hierarchical Phrase-based model Machine Translation

The experiments were conducted to check the applicability of hierarchical phrase-based model in translation between morphologically rich languages and morphologically rich and poor languages. English-Tamil, Malayalam-English pairs of translations were selected for the experiment of translation between morphologically rich and poor language, Tamil and Sinhala languages are chosen for the experiment of translation between morphologically rich languages. Moses 2.0 was used to this research to conduct the experiments. We have used BLEU as our evaluation metric. BLEU considers n-gram overlap between machine translation output and reference translation. Then it computes precision for n-grams of size 1 to 4. It adds brevity penalty for too short translations. The evaluation procedure was carried out on the data mentioned in the above section The goal was to build a machine translation system that can deal with official documents data. Six translation directions were dealt with in the project: Tamil to English, English to Tamil, Malayalam to English, English to Malayalam, Tamil to Sinhala and Sinhala to Tamil.

4.3.1 Baseline system

For the system to compare the results we trained Moses machine translation system on the same data set. This is the simplest machine translation model and is used as a benchmark to compare hierarchical MT system with phrase-based MT system. Default feature set: language model, lexical weighting (both directions), distortion model, word penalty, and phrase penalty are same as hierarchical model. We ran the trainer

with its default settings and then used Koehn's implementation of minimum-error-rate training [15] to tune the feature weights to maximize the system's BLEU score on the development set, yielding the best values. Finally, we ran the decoder on the test set limiting distortions to 4. These are the default settings.

4.3.2 Hierarchical Model

Moses [19] was run on cleaned and preprocessed data using default training scripts. In our work additional switches like hierarchical and glue grammar were also used in training command as the experiments were carried out with the HPB model. The training process begins with a word-aligned corpus. Lmplz [117] was used for the language modeling. 3-gram Language Models (LMs) were created. The featuring weights were tuned using Minimum Error Rate Training (MERT) on 100 best translations [19]. A set of 1500 randomly selected sentences were used for tuning..

Decoding was done using the state-of-the-art Moses using cube pruning techniques with a stack size of 5000 and the maximum phrase length of 5 [118]. The task of decoding in machine translation is to find the best scoring translation according to these formulae. This is a hard problem, since there are an exponential number of choices, given a specific input sentence. In other words, exhaustively examining all possible translations, scoring them, and picking the best is computationally too expensive for an input sentence of even modest length.

The testing phase was completed by using the Moses decoder. The testing was carried out in the same way for all the language pairs. The comparison of the results of HPM SMT and baseline system was conducted. The output of the system was evaluated using Bilingual Evaluation Understudy (BLEU) [11]. The system was evaluated on 500 randomly selected sentences/phrases, where the letter headers and footers were added as comma separated phrases for testing, to ensure that the score of a single sentence no longer depends on a single or very little amount of words.

4.4 POS Integration to SMT system

This section explains the development of factored corpora and integration of Parts of Speech linguistic knowledge in SMT system. We have integrated POS to SMT system to overcome challenges such as reordering, Abbreviations, initials, word flow of the

sentence and mapping one word with one or more word. There are three main components in statistical machine translation system such as,

1) Translation model
2) Language model
3) The Statistical Machine Translation Decoder

POS can be integrated into Translation model and Language model. The first step of integrating POS is the creation of factored corpora with the POS information. The next step is integrating POS factored corpora to the SMT system. The details of those steps are given in following subsections.

4.4.1 Automatic Creation of Factored Corpora

Before providing the bilingual corpus of Sinhala-Tamil language pair and monolingual corpus of Tamil and Sinhala language for creating translation models and language models, both the corpus has to be tokenized in order to separate the words and punctuations i.e., 'தெரிவிக்கின்றேன்.' will be separated as 'தெரிவிக்கின்றேன்' and '.' with space in between them. Tokenization determines where sentence starts and ends. There is a need of cleaning the corpus to remove the sentences from the corpus that exceeds the limit which is the maximum length of the parallel sentences, empty sentences, misaligned sentences and the sentences exceeds 1:9 ratio. Messy and noisy data can disrupt the training process. We need to give both languages at a time as removal of lines should occur concurrently in both languages

As discussed in section 4.1, we have access to the Sinhala-Tamil parallel corpus of government official documents. This parallel corpus is manually cleaned & aligned by three professional translators. This corpus contains more than 24,872 parallel sentences, 1,611,885 monolingual Tamil sentences, and 4,760,531 monolingual Sinhala. This parallel corpus was annotated using the automatic POS tagger of both languages. For the Tamil language, we have used an automatic POS tagger developed by Dhanalakshmi et al of AMRITA University, Coimbatore. The system was trained with a corpus of twenty-five thousand sentences and they claimed accuracy of 95.63% [41]. We have used an automatic POS tagger based on SVM which was developed by the University of Moratuwa, Sri Lanka to annotate the

Sinhala corpus. Researchers reported an overall accuracy of 84.68% [51]. The monolingual corpus of both languages was also annotated using suitable taggers mentioned above. Factored parallel sentences are given in Table 4.3.

Table 8 Factored Parallel Sentences in Sinhala and Tamil

Factored Sinhala Sentence	**Factored Tamil Sentence**
ඒ|DET අනුව|POST එම|DET විභාගයට|NNC ඉල්ලුම්|NCV කරන|VP ලෙස|POST කාරුණිකව|RB දන්වමි|VFM .|FS	அதன்|PR_PRP பிரகாரம்|N_NN அப்|N_NN பரீட்சைக்கு|N_NN விண்ணப்பிக்குமாறு|RB தயவுடன்|N_NN அறியத்தகிறேன்|V_VM_VF .|RD_PUNC
අභ්‍යන්තර|JJ අනුයුක්ත|PCV කිරීම|VNN	உள்ளக|N_NN இணைத்துக்|V_VM_VNF_VBN கொள்ளல்|N_NNP
තාක්ෂණික|JJ ඇගයීම්|NNC කමිටුව|NNC සඳහා|POST නියෝජිතයෙකු|NNC ලබාගැනීම|VNN .|FS	தொழிநுட்ப|N_NN மதிப்பீட்டு|N_NN குழுவிற்கு|N_NN பிரதிநிதியொருவரை|N_NN பெற்றுகொள்ளல்|V_VM_VF .|RD_PUNC

4.4.2 Factored SMT for Sinhala and Tamil Language

Factored translation is an extension of phrase-based statistical machine translation that allows the integration of additional morphological and lexical information, such as lemma, parts of speech, gender, number, etc., at the word level on the source and the target languages. Here we focused on integrating Parts of Speech to SMT system. Figure.4.2 explains the mapping of Sinhala factors and Tamil factors in Factored SMT. Sinhala factors “Lemma” and “POS” are mapped to Tamil factors “Lemma” and “POS”.

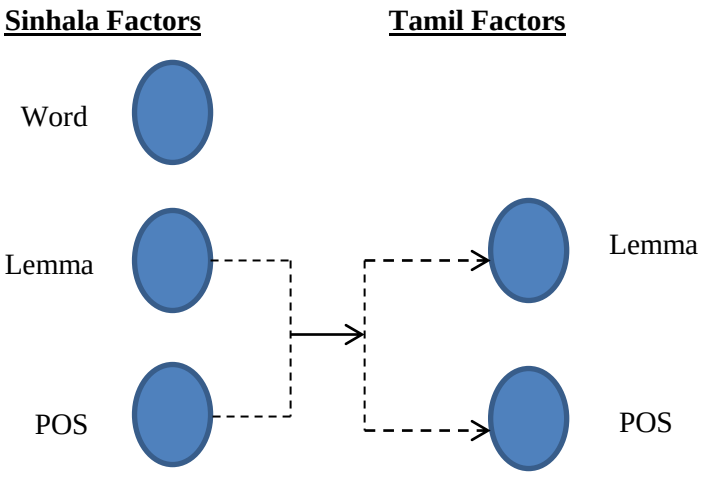


Figure 4.15 Mapping Sinhala factors to Tamil Factors

Three scenarios were tested, which are using the only surface form as a baseline, using POS Tag, and using Google-Translate. Google-Translate was chosen to know how good the results of translation model compared to the legacy machine translator using the phrase-based approach in the official document domain. In the experiment scenario using the POS Tag, we use three kinds of translation model, model 0-0,1; 0,1-0 and 0,1-0,1 and we integrated POS in LM also. The details of these models are depicted in the following table 4.4.

Table 9 Three kinds of translation model and LM

Model	**Description**
0-0,1	PoS tag was added to the source side
0,1-0	PoS tag was added to the target side
0,1-0,1	PoS tag was added to both of the source and Target side and normal LM
0,1-0,1 with tagged LM	PoS tag was added to both of the source and Target side and tagged LM

In the third scenario, the experiment was done by translating the same input text using Google-Translate. The translation result of the Google's is then being evaluated using the same reference text used in the first and second scenario.

Lmplz [117] was used for the language modeling. 3-gram Language Models (LMs) were created. To build a phrase-based translation model, the perl script, 'train-model.perl' in Moses is used. The featuring weights were tuned using Minimum Error Rate Training (MERT) on 100 best translations [19]. A set of 1000 randomly selected sentences were used for tuning. Decoding was done using the state-of-the-art Moses using cube pruning techniques with stack size of 5000 [118]. Figure 4.3 shows the workflow of POS integrated system.

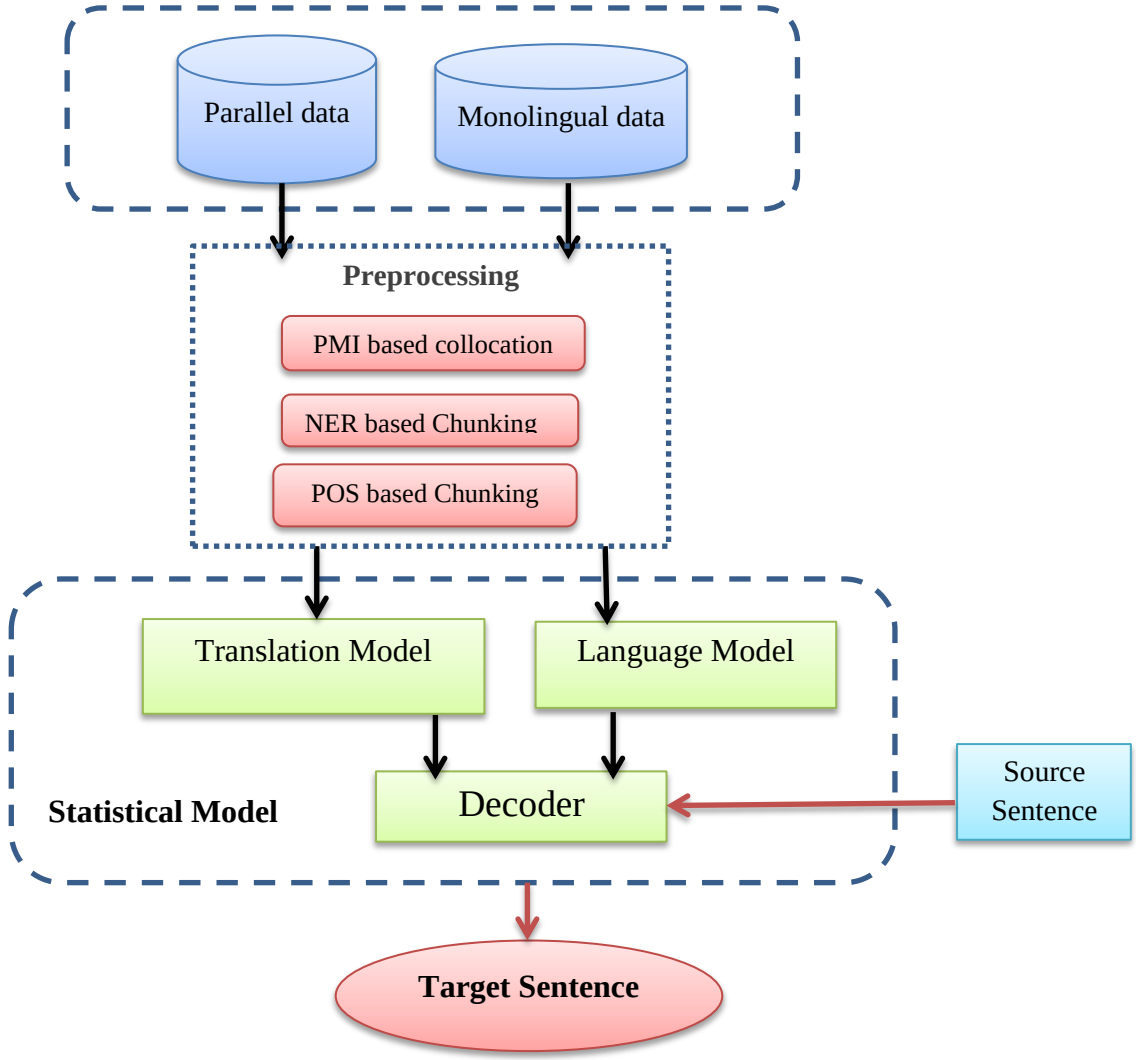


Figure 4.16 Workflow of POS integrated SMT system

The testing phase was completed by using the Moses decoder. A set of 300 sentences are used to test the system. Same test data set is used to test the base line system, POS integrated system and Google translate. The output of the system was evaluated using Bilingual Evaluation Understudy (BLEU) [11].

4.5 Preprocessing based on Chunking

Pre-processing described in the research is related to finding collocation words from PMI, chunking the named entities and POS based chunking. These are described in the following subsections. Figure 4.4 shows the steps of a phrase-based SMT system with pre-processing in experiments.

The bilingual and monolingual data are pre-processed before preparing translation models and language models. These trained models are used by the decoder for translating a given source to target language sentence.

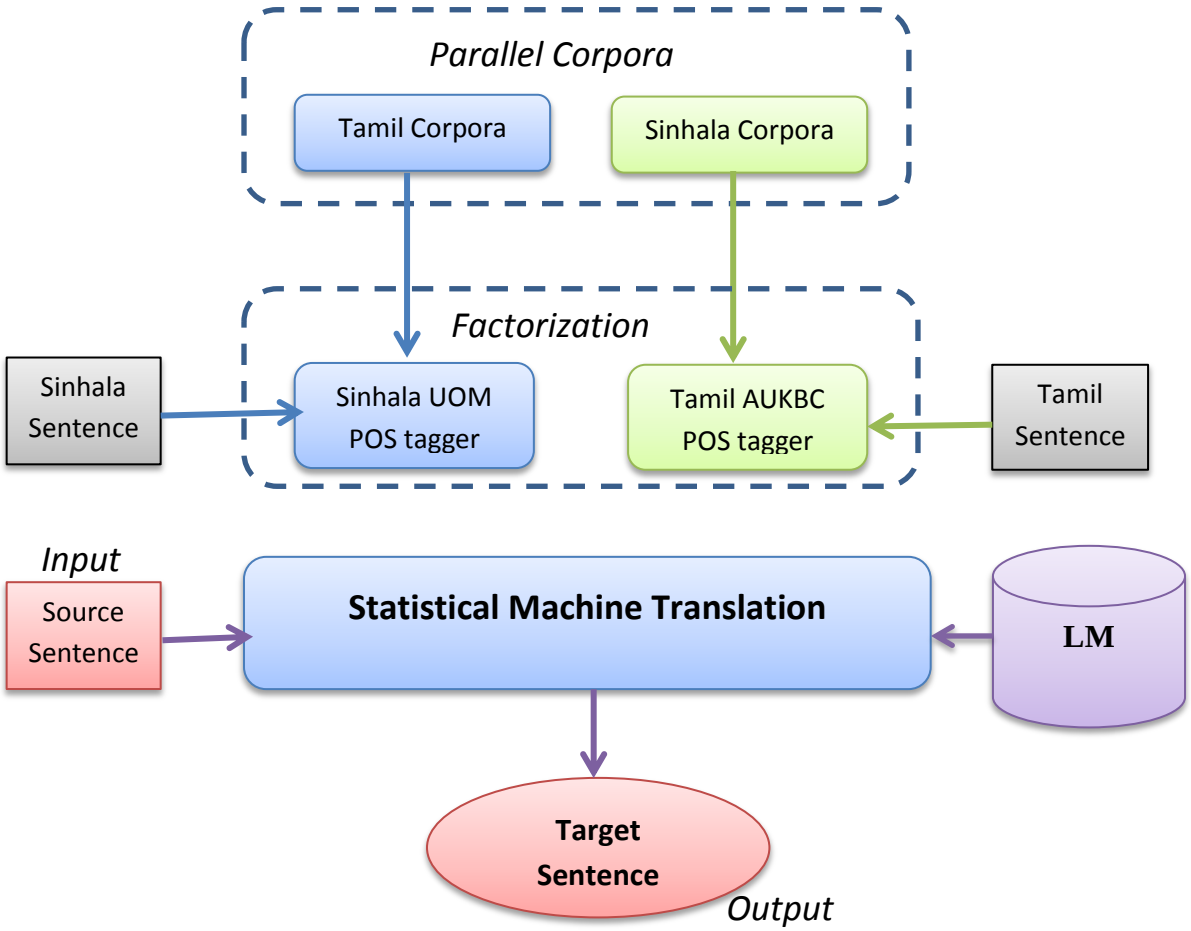


Figure 4.17 Phrase based statistical machine translation system with Preprocessing

4.5.1 PMI based preprocessing

Collocations are expressions of multiple words which commonly co-occur. Collocation extraction is a computational technique that finds collocations in a document or corpus, using various computational linguistics elements resembling data mining [119]. The corpus is pre-processed in such a way that the frequency of co-occurrence word-pair is easily counted from the corpus. The Sinhala and Tamil official document domain corpora used in this approach also. The preprocessed data is used to count the co-occurrence frequency.

The parallel and monolingual data is given to the PMI method and we extracted collocations using no of frequency as 5 and 10. That means if the collocated words occur more than the specified frequency in the corpus only, it will recognize as collocation words. We selected from top 100 to 1000 collocated words when the frequency is 5, based on the PMI score in both languages and we selected from top 500 and 1000 collocated words when the frequency is 10, to evaluate the system. We selected only 500 and 1000 collocated words when the frequency is 10 as they didn't

give a good result like frequency is 5. Then those words are changed as a single word by using an underscore between those words in the parallel corpus.

In the translation training, we utilize the word alignment results using GIZA ++ [15], while in the language model training we use Lmplz [117], which apply the n-gram language model. In the decoding process, we applied Moses Translation System [19], and the BLEU Score as an evaluation method [11]. In this research, we have used preprocessed data only in translation model. The data which is not preprocessed is used for the Language model building. We have trained the system 12 times from top 100 to 1000 collocated words when frequency=5 and 500 to 1000 collocated words when frequency=10.

To compare the results, we trained Moses machine translation system on the same data set, but without preprocessing. This is the simplest machine translation model and is used as a benchmark to compare the results of preprocessed data using PMI score with phrase-based MT system. We ran the trainer with its default settings and then used Koehn's implementation of minimum-error-rate training [15] to tune the feature weights to maximize the system's BLEU score on the development set, yielding the best values. Finally, we ran the decoder on the test set limiting distortions to 4. These are the default settings.

4.5.2 NER based chunking preprocessing

The Sinhala and Tamil official document domain corpora used in this approach also. The pre-processing approach consists of the subsequent steps. Initially, the parallel corpus and monolingual corpus are tokenized and cleaned using the script available with Moses system. As the next step, named entity words are tagged using an automatic NER tagger developed by Mokanarangan et al. of University of Moratuwa, SriLanka. They claimed F-score as 0.82 for Tamil language and 0.79 for Sinhala language for the NER taggers. Bidirectional LSTMCRF model is used to create the NER tagger. As the next step, we have extracted only the name entities belong to person name and addresses.

Then we have run through an algorithm between name entities and parallel corpora to find out the places of those name entities in the parallel corpora. When we find that name entity in parallel corpora, the words are changed as a single word by

using an underscore between those words in the parallel corpus. This preprocessed data was used to train the SMT system.

As the next step, we have given above-preprocessed data to the word alignment tool GIZA ++ [15] to build a translation model. The data which is not preprocessed by NER based chunking is used to build language model using Lmplz [117] tool. So, this research focused on using preprocessing techniques only on translation model, not in the language model. In the decoding process, we applied Moses Translation System [19], and the BLEU Score as an evaluation method [11].

To evaluate the results we trained Moses machine translation system on same data set without preprocessing. This is the simplest machine translation model and is used as a benchmark to compare the results of preprocessed data using NER with phrase-based MT system. We ran the trainer with its default settings and then used Koehn's implementation of minimum-error-rate training [15] to tune the feature weights to maximize the system's BLEU score on the development set, yielding the best values. Finally, we ran the decoder on the test set limiting distortions to 4. These are the default settings.

4.5.2 POS based chunking preprocessing

The Sinhala and Tamil official document domain corpora used in this approach also. The pre-processing approach consists of the succeeding steps. Cleaned parallel corpus was chunked by calling the REST API of POS chunker of both languages. Then we have run through an algorithm between POS based chunks and parallel corpora to find out the places of those POS chunks in the parallel corpora. When we find that POS chunks in parallel corpora, the words are changed as a single word by using an underscore between those words in the parallel corpus. This preprocessed data was used to train the SMT system.

As the next step, translation model is built using above preprocessed data with the help of word alignment tool GIZA ++ [15] along with Moses. The data which is not preprocessed by POS based chunking is used to build language model using Lmplz [117] tool. So, this research focused on using preprocessing techniques based on POS chunking only on translation model, not in the language model. In the

decoding process, we applied Moses Translation System [19], and the BLEU Score as an evaluation method [11].

To evaluate the results we trained Moses machine translation system on same data set without preprocessing. This is the simplest machine translation model and is used as a benchmark to compare the results of preprocessed data using NER with phrase-based MT system. We ran the trainer with its default settings and then used Koehn's implementation of minimum-error-rate training [15] to tune the feature weights to maximize the system's BLEU score on the development set, yielding the best values. Finally, we ran the decoder on the test set limiting distortions to 4. These are the default settings.

After doing all these methods individually, we have experimented hybrid approach by combining all the above three preprocessing techniques.

4.6 Preprocessing based on Segmentation

An unsupervised learning algorithm 'Morfessor' is used to segment the words of the source and target languages in order to train language and translation models in this research. Segmentation means finding morpheme-like units in words. Morfessor Categories-MAP algorithm [18] is used in this research as it gives better segmentation accuracy and handles OOV words in the training process. Words are divided as multiple prefixes followed by stem and multiple suffixes using this algorithm. Some multiple stems are found in some rare cases.

Initially, Sinhala and Tamil corpora are trained separately using Morfessor algorithm and extracted morpheme-like units as shown below.

நடைமுறைப்படுத்த (implemented): நடை முறை ப்படுத்த

ක් රියාත්මක (operating): ක් රි යා ත් ම ක

Then we performed two sets of experiments, one with a word based (Baseline system) and with segmentation for the Sinhala-Tamil language pair.

Baseline System

Standard phrase-based model SMT where words are used as the smallest unit is used in this experiment. This experiment is done to compare the performance against the

segmentation based approach. A sample parallel sentence from the data is shown below.

TA: அந்த புத்தகத்தை அச்சிடுவதற்கு தேவையான ஏற்பாடுகளை ஒழுங்குசெய்து தருமாறு தயவுடன் கேட்டுக்கொள்கிறேன் .

SI: එම ග්‍රන්ථය මුද්‍රණය කිරීමට අවශ්‍ය කටයුතු සලසා දෙන මෙන් කාරුණිකව ඉල්ලා සිටිමි .

The open source SMT system Moses is used with GIZA++ [15] to develop the baseline system. Here we have used standard alignment heuristic grow-diag-final for word alignments. Language models were trained using the Lmplz [117] with 3 –gram. The systems were tuned using a small extracted parallel dataset (1000 sentences) with MERT and after tested with a randomly selected test dataset which contains 300 sentences. Finally, the BLEU evaluation metric was used to evaluate the output produced by the translation system.

Segmentation System

Standard phrase-based model SMT where morpheme-like units are used as the smallest unit is used in this method. The parallel corpora which are segmented as small morpheme-like units are used to train the system. The words in the parallel sentences (training, tuning, testing) and monolingual corpus were replaced with these morpheme-like units. An Example of the split morpheme-like parallel sentence is shown below.

TA: அ ந்த | பு த்த க த்தை | அ ச் சி டு வதற்கு | தேவை யா ன | ஏற்பாடு களை | ஒழுங்கு செய்து| தரு மாறு | த ய வு டன் | கேட் டு க் கொள் கிறேன் |.

SI: එ ම | ග් ර න් ථ ය | මු ද් ර ණ ය | කි රී ම ට | අ ව ශ් ය | ක ට යු තු | ස ල සා | දෙන | මෙන් | කා රු ණි ක ව | ඉ ල් ලා | සි ටි මි |.

Then as mentioned in the baseline system, training, testing, and tuning were done. Finally, the evaluation was done after performing some post-processing. In the post-processing stage, the longest matching morpheme-like units were merged to extract readable translated sentences.

EXPERIMENTS

5.1 Overview

In this section, we present the experimental setup of four different approaches we took towards the improvement of translation quality between Sinhala and Tamil languages. In each experimental setup, we compare the BLEU [11] score of proposed approaches against the baseline system. To come up with the solutions to improve the translation, the challenges in the current machine translation system should be identified. We chose Si-Ta, the state of the art machine translation system for Sinhala -Tamil for this study. As a step towards understanding the translation challenges between Sinhala-Tamil, we studied the divergence between these languages. Divergence helps in defining the possible challenges any machine translation algorithm have to tackle for a given pair of languages. So, the language divergences between Sinhala and Tamil languages are identified and categorized according to the Dorr's classification. Those experiment details are presented in the below section. With the analyzed results of language divergence, a divergence among Tamil and Sinhala POS tagsets could be identified. Accordingly, we have come up with an algorithm for the alignment of different POS tagsets. The details of the semi-automatic alignment also presented in this chapter.

Based on the analyzed results of language divergence, translation challenges such as reordering, abbreviations and initials, word flow of the sentence, data sparseness and mapping one word with one or more words could be found out. Subsequently, we have used hierarchical phrase-based model and Factored model with POS integration to address challenges such as word reordering, word flow, context-aware word selecting, translating conjunction words, better word choice and translating initials and abbreviations. Further, we experimented with few pre-processing techniques based on chunking and segmentation towards addressing challenges such as unknown words, context awareness, better word choice, word flow, ambiguity in translation, translating proper Sandhi, translating name entities and mapping one word with one or more words. PMI based collocation phrases, POS-based chunks, Named Entities and sub word segments are used to enhance the preprocessing step.

This translation system has three processes such as preprocessing, translation and post-processing. So, the results and the errors depend on the preprocessing stages, factored SMT/ hierarchical phrase-based SMT, post-processing and other external modules. All the experiments are carried out by Moses toolkit. The experimental setup, data, installation of SMT toolkit (Moses), training and testing regulations used in the statistical machine translation system are described in below sections. At last, experimental details of Tamil to Sinhala translation also presented in this chapter as Si-Ta system does not have Tamil to Sinhala translation.

5.2 Language Divergence

As the first step to improve the translation among Tamil and Sinhala languages we have done some experiments to find out the divergence between these languages. Translation is a highly arduous task. It targets at conserving semantic and stylistic equivalents of the source text into the target text. When the source sentences are recognized in a different manner in the target language, divergence in translation goes up. Different linguistic and extra-linguistic constraints play pivotal roles in translation resulting in divergences and other issues. Appropriate identification and understanding of these issues are significant in both manual and machine translation.

Dorr's classification is the first approach to classify the divergence between languages. Most of the other researchers also followed Dorr's classification to come up with solutions for their languages. Accordingly, we also have developed this research based on Dorr's classification. Initially, the seven classes of Dorr's classification are studied to get the knowledge of the classes. The results of the language divergence are discussed with three linguistically capable people in both Tamil and Sinhala languages.

The results of the traditional SMT system are used here to identify the translation divergence. Traditional SMT system is built using 24, 872 parallel sentences and 1,611,885 Tamil monolingual sentences in the Sinhala to Tamil direction. 300 sentences are used to test the system.

The results from the testing are analyzed to come up with translation divergence. Then categorization of those findings was held based on the Dorr's classification. According to the Dorr's classification, it has been come up with rules to handle those

divergences. The divergences do not belong to any category proposed by Dorr, we grouped them separately.

Given an input Sinhala sentence and corresponding Tamil sentence, the proposed rules aim at recognizing the occurrence of divergence in the Sinhala to Tamil translation. Let L_{si} is the Sinhala language and L_{ta} is the Tamil language from a tagged bilingual corpus. The following rules are for the identification of lexical-semantic divergence.

Given two languages L_{si} and L_{ta}, a divergence between L_{si} and L_{ta} is a set of correspondences: (x_a, y_α, r) with $x_a \in L_{si}$ and $y_\alpha \in L_{ta}$ being the two matched entities, r being a divergence holding between xa and yα, in this correspondence.

$$x_a : \{ x_a^1 , x_a^2 , ..., x_a^s \}$$

$$y_\alpha : \{ y_\alpha^1 , y_\alpha^2 , ..., y_\alpha^t \}$$

x_a is the sentences from one language and y_α is the sentences from another language. Here in the set of parallel data, L_{si} language has 's' no of sentences and L_{ta} language has 't' no of sentences. There are many possible divergences holding between x_a and

```
1. Input (Tagged bilingual corpus)
2. Repeat step 3 to 11 for each pair of Lsi and Lta, till
the end of the corpus
3. If the main verbs of Lsi and Lta are different
     Then case-I lexical divergence
     //The head of a verb phrase is called as Main verb
```

y_α,

4. If the sentence is an idiom in L_{si}

    Then case-II lexical divergence

5. If the sentence has an Onomatopoeia in L_{ta}

    Then case-III lexical divergence

6. If the adjective of L_{si} is changed into noun of L_{ta}

    Then case-I categorical divergence

7. If the adjective of L_{si} is changed into verb of L_{ta}

    Then case-II categorical divergence

8. If two words in L_{si} is changed to one word in L_{ta}

    Then Inflectional divergence

9. If one word in L_{si} is changed to two word in L_{ta}

    Then Conflational divergence

    Break

10. Else

    No Lexical-Semantic divergence

End of loop

11. Exit

Algorithm 5.1 Algorithm for identifying language divergence Tamil and Sinhala corpus

5.3 Semi-automatic algorithm for aligning different POS tagsets

After finding the language divergence between Tamil and Sinhala languages, the importance of alignment of POS tagsets between Tamil and Sinhala languages was identified. So we proposed a semi-automatic approach for the tagsets alignments. The algorithm was applied on the POS tagsets of Tamil and Sinhala languages as our research is based on improving the efficiency of translation of Tamil and Sinhala translation and we have access to the Sinhala-Tamil parallel corpus of government

official documents. This parallel corpus was manually cleaned & aligned by three professional translators. This corpus contains more than 40,000 words. This parallel corpus was annotated using the automatic POS tagger of both languages. For the Tamil language, we have used an automatic POS tagger developed by Dhanalakshmi et al. [42] of AMRITA University, Coimbatore. The system was trained with a corpus of twenty-five thousand sentences and they claimed accuracy of 95.63% [42]. We have used an automatic POS tagger based on SVM which was developed by the University of Moratuwa, Sri Lanka to annotate the Sinhala corpus. Researchers reported an overall accuracy of 84.68% [51].

Once the annotation was done for both sides of the parallel corpus, the parallel text was word aligned using a word alignment tool. In this study, GIZA++ [15] is used as word alignment tool as it gives higher accuracy for our dataset. GIZA++ can perform word alignments in two directions for each pair of languages by considering one language as source and other as the target. The intersection of both directions is taken as the resulting alignment.

In order to proceed with tagset alignment, initially, a number of words belong to each tag was calculated in both language which resulted in most of the words into "common noun" category. Based on the word alignment, a tag alignment was retrieved. This resulted in any tag of one language can be mapped to any tag of the other. In our study, there are 35 tags from BIS tagset and 30 tags from UOM tagset. So there can be 30*35 (1050) possible alignments of tags. Further to refine this alignment, statistical values of this mapping was considered. The highest three mappings were considered as the possible aligned tags. The highest three mappings were derived using an automatic program by counting words belongs to each mapping.

The general idea is to consider all the tag alignments of both languages that were generated from the GIZA++ algorithm and chose the most frequent of them as the correct alignment. But, in our approach, we chose top three frequent aligned tags and cross-checked it with bilingual experts to finalize the alignments. For example "Nipathana" in UOM tag aligned with "Verb Finite" and "Common noun" mostly in BIS tagset. But through the linguistic point of view, it should have to align with "Verb finite".

5.4 Hierarchical phrase based machine translation

Hierarchical phrase-based model is used to overcome the challenge of quality of the translation. This section discusses the training, tuning, and testing of different model components. The evaluation was carried out on Ubuntu 16.00 running on Intel Core i5 machine with 2GB of RAM and 500GB of Hard disk space. The experiments were conducted to check the applicability of hierarchical phrase-based model in translation between morphologically rich languages and morphologically rich and poor languages. English-Tamil, Malayalam-English pair of translations were selected for the experiment of translation between morphologically rich and poor language, Tamil and Sinhala languages are chosen for the experiment of translation between morphologically rich languages. The subsections discuss the data set used in this research and the experimental setup.

5.4.1 Dataset

We used the IIIT-Hyderabad (International Institute of Information Technology) parallel corpus for Tamil-English and Malayalam-English languages. They have corpora of eleven languages. Size of each corpus is about 3 million words. Texts in each corpus are categorized under aesthetics, mass media, social science, natural science, commerce and translated materials. The corpora were prepared by several organizations under the funding from MoIT (Ministry of Information Technology formerly Department of Electronics), Government of India. Its bilingual resources consist of roughly about 50,000 sentences for all the available languages [37]. The corpora are already sentence aligned. Here we have cleaned this corpus for making it completely compatible.

The main source of the parallel corpus of Sinhala-Tamil languages is government official documents. The documents collected from government institutions were hard copies and some were of a single source. They are generally translated manually with the aid of human translators. We digitalized those written documents into text files by crowdsourcing. The typed documents were sentence aligned with the manual inference. Its bilingual resources consist of about 22,000 sentences for Tamil and Sinhala languages. Further details about parallel data are given in Table 5.1. The target language corpus in above parallel corpus is used in the development of language model for this study work.

Table 10 Complete Statistics of Parallel Corpus (In Sentence)

	Tamil-English	**Malayalam-English**	**Tamil-Sinhala**
Training	48,000	48,000	20,000
Tuning	1,500	1,500	1,500
Testing	500	500	500

5.4.2 Experimental setup

As the initial step of the experiments, the obtained data was tokenized using customized scripts and standard Moses [71] filtration was utilized to confirm that the sentences with an extreme length ratio difference were removed effectively. English language corpus was followed by lowercasing by the script being supplied with the Moses decoder [19]. This training data was used for word alignment. Moses was run using Koehn's training scripts. In our work additional switches like hierarchical and glue grammar also used in training command as the experiments were carried out with the HPB model.

For the other parameters, the default values were used i.e. 3-gram language model and maximum phrase length= 6. Giza++ [15] was used for the word alignment with 'grow-diag-final-and' as the summarization heuristics. Lmplz [117] was used for the language modeling. 3-gram Language Models (LMs) were created. The featuring weights were tuned using Minimum Error Rate Training (MERT) on 100 best translations. A set of 1500 randomly selected sentences were used for tuning. Decoding was done using the state-of-the-art Moses using cube pruning techniques with a stack size of 5000 and the maximum phrase length of 5. The testing phase was completed by using the Moses decoder. The testing was carried out in the same way for all the language pairs. For the comparison of the results of HPM SMT, we have done traditional SMT approach also to the same data set. Traditional SMT approach for the same data set also used for the comparison of the HPM SMT results.

The output of the system was evaluated using Bilingual Evaluation Understudy (BLEU) [11]. The system was evaluated on 500 randomly selected sentences/phrases, where the letter headers and footers were added as comma separated phrases for testing, to ensure that the score of a single sentence no longer depends on a single or very little amount of words.

As hierarchical phrase-based model did not give a good result for the Sinhala to Tamil translation, we have moved to POS integration to solve the reordering issue.

5.5 POS Integration to SMT system

As another way of enhancing the quality of the translation, Factored model with POS integration is adopted in this research. Training, tuning, and testing of different model components of the factored model are discussed in this section. The evaluation was carried out on Ubuntu 16.00 running on Intel Core i5 machine with 2GB of RAM and 500GB of Hard disk space. The experiments were conducted to check the applicability of factored model in translation in the direction of Sinhala to Tamil translation. The subsections discuss the data set used in this research and the experimental setup.

5.5.1 Dataset

In order to develop this system, the data is collected from different sources basically in the areas that are more related to the official documents domain. Gathered data was classified into two based on the context and writing style such as in-domain and out-domain. Data gathered from official letters (e.g., from the Department of Education, Administrative department etc.) and additional data from other government sources such as annual reports, parliament order papers, circulars and establishment codes were considered as in-domain. Even though these were from government organizations, the writing style was diverse from official letters described above (e.g. the parliament order papers were more like question and answer form).

Some source documents of in-domain were hard copies in a single language (either the Tamil or Sinhala version of the document), while some were soft copies in PDF format. The single-language source documents were manually translated and typed. A custom developed tool was used to extract data from PDF documents. The sentence alignment tool created by Hameed et al. [122] was used to create the parallel data. A custom script is used to make sure that there are no duplicates in the training, tuning and testing sets. In addition, we collected some monolingual Tamil sentences of this category from the annual reports.

Other easily accessible data sources were from the web, (such as articles from blogs, news, and wiki dumps), and other free sources. This out-domain data was collected from some freely available sources (Ramasamy et al., 2012, Goldhahn et al., 2012, IIIT-Hyderabad, Tamil news crawl, Tamil Wikipedia, Fire corpus) as well as

gathered via web crawling and getting access from the owner. So far, the context, with respect to official government letters was fairly dissimilar. Therefore, these were classified as out-domain data. However, it is possible only to gather monolingual data under this category.

The test set was prepared for evaluations. The test set is a set of sentences randomly picked from the collection from where the training and tuning data were derived. The average sentence lengths of test set were 10.95 and 9.90 for Sinhala and Tamil, respectively. Statistics on the parallel data and Tamil monolingual data are shown in Table 5.2, and Table 5.3, respectively.

Table 11 Sources of parallel data

Source	**Sentences**	**Words (Sinhala)**	**Words (Tamil)**
In-domain	9,227	79, 407	71,407
Pseudo in-domain	15,645	237,498	197,271
Tuning	1,000	12,441	10,641
Test set	300	4,015	3,394

Table 12 Tamil Monolingual Data

Source	**Sentences**	**Words (Tamil)**
In-domain	9,227	71,407
Pseudo in-domain	76,692	788,544
Out-domain	1,525,966	21,348,157
Total	**1,611,885**	

5.5.2 Experimental setup

Tokenizing obtained data using customized scripts is the initial step of this experiment. After tokenizing the data, standard Moses [19] filtration script was utilized to remove the sentences with extreme length ratio, blank sentences, misaligned sentences and extremely large sentences. This parallel corpus was annotated using the automatic POS tagger of both languages. For the Tamil language, we have used an automatic POS tagger developed by Dhanalakshmi et al of AMRITA University, Coimbatore. The system was trained with a corpus of twenty-five

thousand sentences and they claimed accuracy of 95.63% [42]. We have used an automatic POS tagger based on SVM which was developed by the University of Moratuwa, Sri Lanka to annotate the Sinhala corpus. Researchers reported an overall accuracy of 84.68% [51]. The monolingual corpus of Tamil language also annotated using suitable taggers mentioned above. Example of annotated corpora is shown below.

E.g

තාක්ෂණික|JJ ඇගයීම්|NNC කමිටුව|NNC සඳහා|POST නියෝජිතයෙකු|NNC ලබාගැනීම|VNN .|FS

தொழிநுட்ப|N_NN மதிப்பீட்டு|N_NN குழுவிற்கு|N_NN பிரதிநிதியொருவரை|N_NN பெற்றுகொள்ளல்|V_VM_VF .|RD_PUNC

Three scenarios were tested, which are using the only surface form as a baseline (traditional SMT), using POS Tag and using Google-Translate. In the experiment scenario using the POS Tag, we use three kinds of translation model, such as model 0-0,1; 0,1-0 and 0,1-0,1 and we integrated POS in LM also. The details of these models are depicted in the following table 5.4.

Table 5.13 Three kinds of translation model and LM in POS integration

Model	**Description**
0-0,1	Adding POS tag to source side
0,1-0	Adding POS tag to target side
0,1-0,1	Adding POS to both source and target side and normal LM
0,1-0,1 with tagged LM	Adding POS to both source and target side and tagged LM

Five different types of model are trained, tuned and tested with the help of parallel corpora. The general categories of the models are Baseline and Factored systems. The detailed models are,

1. Baseline (BL)
2. Factored system with adding POS tag to the source side + normal LM (SoPOS)
3. Factored system with adding POS tag to the target side + normal LM (TaPOS)

4. Factored system with adding POS tag to the source side + POS tag to the target side + normal LM (SoPOS+TaPOS)
5. Factored system with adding POS tag to the source side + POS tag to the target side + factored LM (SoPOS+TaPOS+factLM)

We have translated the same input text using Google-Translate as the third experiment scenario. The result is evaluated by using same reference text which is used in first and second scenarios.

In the translation training, we have utilized the word alignment results using GIZA ++ [15], while in the language model training we used Lmplz [117], which apply the n-gram language model. In the decoding process, we applied Moses Translation System [19], and the BLEU Score as an evaluation method [11].

Baseline system is a traditional phrase based system. It is built using surface forms of the word. We have used 3-gram language model and Moses as the decoder. Cleaned raw parallel corpus is used for training the system. Lexicalized reordering model (msd-bidirectional-fe) was used in the baseline with automatic reordering model. For factored model, instead of using the surface form of the word and POS tags are included into the word as additional factors. A factored parallel corpus is used for training the system.

Lmplz [117] was used for the language modeling. 3-gram Language Models (LMs) were created. For the other parameters, the default values were used i.e. 3-gram language model and maximum phrase length= 6. Giza++ was used for the word alignment with 'grow-diag-final-and' as the summarization heuristics. To build a phrase-based translation model, the perl script, 'train-model.perl' in Moses was used. But for factored model training, another parameter called '-translation-factors' need to be added. The values for this parameter differ according to the model. The values according to models are given below.

- SoPOS: 0-0,1
- TaPOS: 0,1-0
- SoPOS+TaPOS: 0,1-0,1
- SoPOS+TaPOS+factLM: 0,1-0,1 and need to specify POS LM and surface word LM

POS LM was built based on only tags of target language without specifying words.

Surface LM was built based on only words. For factored based translation model building, we do not need to put the parameter –'reordering' with the value of 'msd-bidirectional-fe' because in the factored model, default reordering feature is not used. Reordering based on POS happens in the factored model.

The featuring weights were tuned using Minimum Error Rate Training (MERT) on 100 best translations. A set of 1000 randomly selected sentences were used for tuning. Decoding was done using the state-of-the-art Moses using cube pruning techniques with stack size of 5000 and the maximum phrase length of 5. The testing phase was completed by using the Moses decoder. Baseline model is used to compare the effectiveness of factored mode in the translation.

The output of the system was evaluated using Bilingual Evaluation Understudy (BLEU). The system was evaluated on 300 randomly selected sentences/phrases. Same test data set is used to test the baseline system, POS integrated system with all models and Google translate.

In addition to the BLEU scores, two human evaluations were used to verify the applicability/usability of the translations. Evaluation is done on the output of the first 300 test sentences. In the human evaluation, more precedent is given to the word error rate and on the context as a whole. This helped to normalize the issues of n-gram matching in BLEU scoring, as well to evaluate the translation based on overall accuracy, fluency and usability considering the context.

5.6 Preprocessing based on chunking

Preprocessing is another way of enhancing the quality of the translation. Preprocessing described in the research is related to finding collocation words from PMI, NER based chunking, and POS based chunking. Details of above three preprocessing techniques are described in the following subsections individually. Training, tuning, and testing of different above approaches are discussed in this section. The evaluation was carried out on Ubuntu 16.00 running on Intel Core i5 machine with 2GB of RAM and 500GB of Hard disk space. The experiments were in the direction of Sinhala to Tamil translation. The subsections discuss the data set used in this research and the experimental setup.

5.6.1 Dataset

As discussed in the section 5.5.1, we have access to the Sinhala-Tamil parallel corpus of government official documents. This parallel corpus was manually cleaned & aligned by three professional translators. This corpus contains more than 24,872 parallel sentences and 1,611,885 monolingual Tamil sentences. Statistics of training, tuning, testing, and language model are shown in table 5.5. The test set was prepared for evaluations. The test set was a set of sentences randomly picked from the collection from where the training and tuning data were derived. The average sentence lengths of test set were 10.95 and 9.90 for Sinhala and Tamil, respectively.

Table 14 Statistics of training, tuning, testing and language model

	No of sentences in Sinhala	**No of sentences in Tamil**
Training	24,872	24,872
Tuning	1,000	1,000
Testing	300	300
Language model	4,760,531	1,611,885

5.6.2 Experimental setup for PMI based chunking

In this experiment, we found out collocation words using Point-wise Mutual Information (PMI) technique. Collocations are expressions of multiple words which commonly co-occur. The corpus was pre-processed in such a way that the frequency of co-occurrence word-pair is easily counted from the corpus and chunked those words. The Sinhala and Tamil corpora mentioned above used in this approach.

The preprocessing approach consists of the following steps. Initially, the parallel corpus and monolingual corpus were tokenized using a customized script for Tamil and Sinhala languages. After that, the parallel corpus was cleaned using the script available with Moses system to remove misaligned sentences in the corpus. Then, special characters including numeric digits and full stops were removed from the corpus. It is noticed that if any special character is present in between any two consecutive words then they are not considered as a co-occurrence word-pair to extract collocation. Therefore, after removal of any special character, the line was broken to extract collocation properly. Then the data was used to count the co-

occurrence frequency. The collocation words found out using a PMI based algorithm using NLTK library in python. The PMI algorithm is shown below.

Import NLTK library

Open the file

For each line in the file

    Read the line in encoding utf-08 format

    Split the sentences into word

Initialize bigram object

Get all bigrams in the corpus using PMI method

Filtering the frequent bigrams (in 2 scenarios: more than 5 and 10)

Put frequent bigrams in PMI measured list

Close the file

Open a new file to write

FOR each bigrams in PMI measured list

    Write it into a file

END FOR

Close the file

Algorithm 5.2 Collocation finding algorithm using PMI

The parallel and monolingual data was given to the PMI method and we extracted collocations using no of frequency as 5 and 10. That means if the collocated words occur more than the specified frequency in the corpus only, it will recognize as collocation words. It is often important to remove low-frequency candidates, as we lack sufficient evidence about their significance as collocations. Here we have used bigram measurement as we have focused on two consecutive words. For Sinhala and Tamil languages when the frequency is 5, we got 9254 and 6379 collocated words respectively. For Tamil and Sinhala languages when the frequency is 5, we got 3818 and 2521 collocated words respectively. So we selected from top 100 to 1000 collocated words when the frequency is 5, based on the PMI score in both languages

and we selected from top 500 and 1000 collocated words when the frequency is 10, to evaluate the system. We selected only 500 and 1000 collocated words when the frequency is 10 as they didn't give a good result like frequency is 5. Then algorithm 5.3 is used to find out the location of selected bigrams in the parallel corpus. After finding the location of the bigrams in the corpus, those words are changed as a single word by using an underscore between those words in the parallel corpus. 'ஒட்டுமொத்தப்' and 'பொருளாதாரத்திற்கும்' words are changed to 'ஒட்டுமொத்தப்_பொருளாதாரத்திற்கும்'. Likewise Sinhala words 'සමාජ' and 'ඒකාබද්ධතා,' words are changed to 'සමාජ_ඒකාබද්ධතා'. This preprocessed data was used to train the SMT system.

```
Get the Bigram list

Open the corpus

FOR each line in the file

    Read the line

  FOR each bigrams in the list

        IF line contains bigram

            Replace words by adding an underscore between those words

    END FOR
    Write the line in a new file

END FOR
```

Algorithm 5.3 Find out the location of selected bigrams in the parallel corpus

In the translation training, we utilize the word alignment results using GIZA ++ [15], while in the language model training we use Lmplz [117], which apply the n-gram language model. Lexicalized reordering model (msd-bidirectional-fe) was used in the system with automatic reordering model in the training processes. 1000 sentences were used to tune the system. 300 sentences were used to test the system. In the decoding process, we applied Moses Translation System [19], and the BLEU Score as an evaluation method. We have trained the system 12 times from top 100 to

1000 collocated words when frequency=5 and 500 to 1000 collocated words when frequency=10 and compared the result to find best frequency point.

5.6.3 Experimental setup for NER based chunking

A named entity is a real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name. Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction that seeks to locate and classify named entities in text into predefined categories [119]. In this experiment, we found out named entity words in the corpus and combined it into a single word. But in this research, name entities belonged to person name and addresses are also considered. Other name entities are not considered in this research. The Sinhala and Tamil corpora mentioned above used in this approach.

Tokenizing the obtained data using customized scripts is the initial step of this experiment. After tokenizing the data, standard Moses [19] filtration script was utilized to remove the sentences with extreme length ratio, blank sentences, misaligned sentences and extremely large sentences. This parallel corpus was annotated by calling the REST API of NER tagger of both languages. For the Tamil and Sinhala languages, we have used an automatic NER tagger developed by Mokanarangan et al. of University of Moratuwa, Sri Lanka. The system was trained with a corpus of 24,872 sentences in both languages and they claimed F-score as 0.82 for Tamil language and 0.79 for Sinhala language. Bidirectional LSTMCRF model is used to create the NER tagger. So accuracy on the output depends on NER tagger as well. Example of annotated corpora is shown below.

E.g

இளங்கோ: B-Person, தனுராஜ்: B-Person

ඉලන්කෝ: B-Person තනුරාජ්: B-Person

In this example, these two words need to merge using underscore as both names specify a person. Then algorithm 6.4 was used to find out the location of those words which occurs consequently in the parallel corpus. After finding the location of the named entity in the corpus, those words were changed as a single word by using an underscore between those words in the parallel corpus. ‘இளங்கோ’ and ‘தனுராஜ்’

words were changed to ‘இளங்கோ_தனுராஜ்’. Likewise Sinhala words ‘ඉලන්කෝ’ and ‘තනුරාජ්’ words were changed to ‘ඉලන්කෝ _ තනුරාජ්’. This preprocessed data was used to train the SMT system.

Get the NER list

Open the corpus

FOR each line in the file

Read the line

FOR each Named Entity in the list

IF line contains Name Entity

Replace words by adding an underscore between those words

END FOR

Write the line in a new file

END FOR

Algorithm 5.4 Find out the location of the named entity in the parallel corpus

Lmplz [117] was used for the language modeling. 3-gram Language Models (LMs) were created. GIZA++ was used to align the words between the source and target languages. When training the system, lexicalized reordering model (msd-bidirectional-fe) was used in the system. 1000 sentences were used to tune the system. 300 sentences were used to test the system. In the decoding process, we applied Moses Translation System, and the BLEU Score as an evaluation method. The result was compared with baseline system.

5.6.4 Experimental setup for POS based chunking

Chunking is a process of extracting phrases from unstructured text. Instead of just simple tokens which may not represent the actual meaning of the text, it is advisable to use phrases such as **South Africa**” as a single word instead of ‘**South**’ and ‘**Africa**’ separate words. Chunking works on top of POS tagging, it uses POS-tags as input and provides chunks as output. Similar to POS tags, there are standard set of Chunk tags like Noun Phrase (NP), Verb Phrase (VP), etc. Chunking is very important when you

want to extract information from text. In this experiment, we found out chunked words based on the POS in the corpus and combined them into a single word. The Sinhala and Tamil corpora mentioned in subsection 6.6.1 used in this approach.

As the initial step, the parallel and monolingual data was tokenized using the customized script and the parallel corpus is cleaned using standard Moses filtration script. Filtration script was utilized to remove the sentences with extreme length ratio, blank sentences, misaligned sentences and extremely large sentences. This parallel corpus was chunked by calling the REST API of POS chunker of both languages. For the Tamil language, we have used an automatic POS Chunker developed by Mokanarangan et al. of University of Moratuwa, SriLanka. The system was trained using Tamil FIRE corpus with an eighty thousand word in Tamil language and they claimed F-score as 0.8 for Tamil language Bidirectional LSTMCRF model is used to create the Tamil POS Chunker. As there is no available chunker for Sinhala language, a freely available CRF based chunker for English language which modified by the Sinhala training data is used in this approach. Example of chunked corpora is shown below.

E.g:

B-NP பணிப்பாளர் B-NP நாயகம் I-NP ,

අධ්‍යක්ෂ/NNC/B-NP ජනරාල්/NNC/B-NP ,/PUNC/B-NP

In this example, these two words need to merge using underscore to preprocess the data based POS chunking. Then algorithm 6.5 was used to find out the location of those words which occurs consequently in the parallel corpus. After finding the location of the chunk in the corpus, those words were changed as a single word by using an underscore between those words in the parallel corpus. 'பணிப்பாளர்' and 'நாயகம்' words were changed to 'பணிப்பாளர்_நாயகம்'. Likewise Sinhala words 'අධ්‍යක්ෂ' and 'ජනරාල්' words were changed to 'අධ්‍යක්ෂ_ ජනරාල්'. This preprocessed data was used to train the SMT system.

Get the Chunk list

Open the corpus

FOR each line in the file

Read the line

FOR each chunk in the list

IF line contains chunk & next word's tag are equal

Replace words by adding an underscore between those words

END FOR

Write the line in a new file

END FOR

Algorithm 5.5 Find out the location of POS chunk in the parallel corpus

To create language model Lmplz was used with 3-gram. GIZA++ was used to align the words between the source and target languages. When training the system, lexicalized reordering model (msd-bidirectional-fe) was used in the system. 1000 sentences were used to tune the system. 300 sentences were used to test the system. In the decoding process, we applied Moses Translation System, and the BLEU Score as an evaluation method. The result was compared with baseline system.

After doing all these methods individually, we have experimented hybrid approach by combining all the above three preprocessing techniques.

5.7 Preprocessing based on segmentation

To improve the efficiency of the translation, preprocessing by segmenting the subwords is chosen as another way. Morphological information in target and source languages is integrated to SMT in this approach. We have conducted our experiments for the Sinhala-Tamil language pair. Training, tuning, and testing of different above approaches are discussed in this section. The evaluation was carried out on Ubuntu 16.00 running on Intel Core i5 machine with 2GB of RAM and 500GB of Hard disk space. The experiments were in the direction of Sinhala to Tamil translation. The subsections discuss the data set used in this research and the experimental setup.

5.7.1 Dataset

The training data consists of 22,872 Sinhala and Tamil parallel sentences in Official document domain. Sinhala-Tamil parallel corpus of government official document (University of Moratuwa, Project funded by Department of Official Languages) is used in experiments. This parallel corpus was manually cleaned &

aligned by three professional translators. The training set was built with 22,872 parallel sentences and a test set was constructed with 300 sentences. 1000 parallel sentences were used for tuning the system. For language model, sizes of 1,611,885 Tamil sentences were used. The average sentence lengths of test set are 10.95 and 9.90 for Sinhala and Tamil, respectively. Statistics of training, tuning, testing, and language model are shown in table 5.6. The test set was prepared for evaluations.

Table 15 Statistics of training, tuning, testing and language model

	No of sentences in Sinhala	**No of sentences in Tamil**
Training	24,872	24,872
Tuning	1,000	1,000
Testing	300	300
Language model	4,760,531	1,611,885

5.7.2 Experimental setup for segmenting the words into sub-word

All the experiments were done in the Sinhala to Tamil translation direction. Fully morpheme-like segmentation is done repeatedly for two different language models (3-gram and 7-gram) without changing the default phrase length. The word-based base-line approach was carried out only for the default settings (i.e. phrase length: 7and 3-gram language model).

As the initial step, the parallel and monolingual data were tokenized using the customized script and the parallel corpus was cleaned using standard Moses filtration script. Filtration script was utilized to remove the sentences with extreme length ratio, blank sentences, misaligned sentences and extremely large sentences. This parallel corpus was segmented by using Morfessor Algorithm [18], an unsupervised learning algorithm, to find morpheme-like units of the source and target languages in order to train the language and translation models. Since Morfessor Categories-MAP algorithm has better segmentation accuracy and handles OOV words in the training data, we have used it in our work.

In the translation training, we utilized the word alignment results using GIZA ++ [15], while in the language model training we used Lmplz [117], which apply the n-gram language model. Here we have used 3-gam and 7-gram language model for the same dataset. Lexicalized reordering model (msd-bidirectional-fe) was used in the

system with automatic reordering model in the training processes. 1000 sentences were used to tune the system. 300 sentences were used to test the system. In the decoding process, we applied Moses Translation System, and the BLEU Score as an evaluation method. We have trained the system 12 times from top 100 to 1000 collocated words when frequency=5 and 500 to 1000 collocated words when frequency=10 and compared the result to find best frequency point. After decoding the sentence from source language to target language, we have used some post-processing technique to merge sub-words into words.

5.8 Tamil to Sinhala traditional SMT system

This research is the extension work of the ongoing project Si-Ta in the University of Moratuwa which is funded by Department of the Official project. "Sinhala to Tamil translation" is already available in the system. So this research focused on developing "Tamil to Sinhala translation". Below subsections discuss the data used in this research and experiments.

5.8.1 Dataset

In order to develop this system, the data is collected from different sources basically in the areas that are more related to the domain. The sources of parallel corpora are the government official documents. The government official documents were collected from various government institutions. With the help of human translators, they were manually translated. They were digitalized (handwritten ones were typed into text files) by crowdsourcing. With the manual intervention, the typed documents were sentence aligned. Table.1 summarizes the statistics on the data used for parallel corpus creation. Therefore average length sizes of this set of letters are lower than other data sources.

Other than the target side of the parallel corpus, the language model was expanded by adding out of domain data from different sources. Other easily accessible data sources were from the web, (such as articles from blogs, news, and wiki dumps), and other free sources. This data was collected from some freely available sources as well as by web crawling. Yet, their context with respect to official government letters was quite different. Therefore, these were categorized as out-domain data. However, it was possible only to gather monolingual data under this category. Statistics on the

parallel data and Sinhala monolingual data are shown in Table-5.7, and Table 5.8, respectively.

Table 16 Sources of parallel data

Source	**Sentences**	**Words (Sinhala)**	**Words (Tamil)**
In-domain	9,227	79, 407	71,407
Pseudo in-domain	15,645	237,498	197,271
Tuning	1,000	12,441	10,641
Test set	300	4,015	3,394

Table 17 Sinhala Monolingual Data

Source	**Sentences**	**Words (Sinhala)**
In-domain	9,227	79, 407
Pseudo in-domain	15,646	237,498
Out-domain	4,735,658	72,531,342
Total	**4,760,531**	

5.8.2 Experimental setup for Tamil to Sinhala SMT

The data was tokenized using customized scripts and standard Moses filtration was utilized to confirm that the sentences with an extreme length ratio difference are removed effectively. The Phrase-Based Statistical Machine Translation (PBSMT) was conducted based on the log-linear model. Giza++ [17] was used for the word alignment with *'grow-diag-final-and'* as the summarization heuristics and *'msd-bidirectional-fe'* as the reordering technique.

Lmplz [117] was used for the language modeling. 3-gram Language Models (LMs) were created with back-off and modified Kneser-Ney smoothing as smoothing technique. The featuring weights were tuned using Minimum Error Rate Training (MERT) on 100 best translations. And a set of 1000 randomly selected sentence was used for tuning. De-coding was done using the state-of-art Moses using cube pruning techniques with a stack size of 5000 and the maximum phrase length of 5.

The output of the system was evaluated using Bilingual Evaluation Understudy (BLEU). The system was evaluated on 300 randomly selected sentences/phrases, where the letter headers and footers were added as comma separated phrases for

testing, to ensure that the score of a single sentence no longer depends on a single or very little amount of words.

RESULTS AND DISCUSSION

To complete this study properly, it is necessary to analyze the results obtained in order to check the objective of this research has accomplished. The present study was an attempt to enhance the quality of Sinhala to Tamil translation with regards to the problems identified by analyzing the results of language divergence. The results obtained were carried out through statistical analysis and are presented in this chapter. For better understanding, the results are divided and presented as following seven heads.

The first section presents the results of language divergence in the Sinhala to Tamil translation. The second section presents the alignment between Sinhala and Tamil POS tagsets using the semi-automatic algorithm. The third section contains the results and comparisons of hierarchical phrase-based model with baseline system. The fourth section contains the results and comparisons of the factored model with baseline system. The fifth and sixth sections present the result of preprocessing techniques based on chunking and segmentation and compared the results with baseline system. The last section presents a translation of Tamil to Sinhala language.

6.1 Language Divergence

This section describes the types of divergence between Tamil and Sinhala in detail. Dorr has classified lexical-semantic divergence into seven types for English into Spanish and English into German translation. These types are Thematic Divergence, Promotional Divergence, Demotional Divergence, Structural Divergence, Lexical Divergence, Categorical Divergence, and Conflational Divergence. Taking this classification, 4 types out of 7 are identified for Sinhala-to-Tamil translation. Some additional divergence types which are not fall under the Dorr's classification are also identified for Sinhala-to-Tamil translations. Below we have presented types of divergence individually.

6.1.1 Conflational Divergence

Conflational divergence is mainly concerned with the verb of the SL. This divergence occurs when a single word in SL requires at least two words of TL to represent the translation. Such type of divergence is found in Sinhala-to-Tamil translation. For example

ඇය නටයි

அவள் நடனம் ஆடினாள்

She danced

In this example, the නටයි Sinhala sentence requires two words of Tamil (i.e. நடனம் ஆடினாள்) upon translation.

There are some nouns also like this. For example,

ආකල්පය -**பொதுமை நோக்கு** – concept

ක්‍රීඩාංගනය -**விளையாட்டு மைதானம்**-Play ground

6.1.2 Inflectional Divergence

Inflectional divergence is the reverse case of conflational divergence. This divergence occurs when two or more than two words of SL require one word of TL. In many cases, the verb of Sinhala language, which contains noun attached to it, is equivalent to one word verb of Tamil language. For example,

ඇය පාඩම් කරනවා

அவள் படிக்கிறாள்

She studies

In this example, the verb පාඩම් කරනවා of Sinhala sentence is equivalent to one word verb of Tamil (i.e. **படிக்கிறாள்**) upon translation. Further examples can be denoted as follows.

දේශපාලන විද්‍යාව- **அரசியல்** - Political science

රජ මාළිගය- **அரண்மனை**-Palace

කියනු ලබන දෙය-**என்பது** -The thing that is said

ඒ කුමක් නිසාද යත්- **ஏன் என்றால்** - That is because of

Most of the compound words including compound nouns and compound verbs mostly tend to be two or more words in Sinhala language while they appear as one word in Tamil language. In translation, this fact emerges as the main issue in mapping words as well as in identifying the correct meaning. As an example, the compound

verb ‘වී ඇත’(has happened) is composed of two words and both words have separate meanings in Sinhala language. The first word ‘වී’ is having the meaning of நெல் (paddy) and ‘ඇත’ consists with the meaning of இருக்கிறது (have). The direct translation for the compound verb should be ‘செயற்படுத்தப்பட்டுள்ளது’ in target language. While considering the above compound verb as separate two words and selecting the separate meanings in translation, the translation in target language appears as . In such cases, correct meaning of the compound words comes with an issue. As a result, such words should be considered as one token, the translation should be aligned with it and the divergence should be considered.

6.1.3 Categorical Divergence

Categorical divergences are located in the mismatch between parts of speech of the pair of translation languages. In case of categorical divergence, changes are in the category. For example, the adjective in one language can be considered as a common noun in another language. For example, in a noun phrase like “පාසල් ලත්ත” (school garden), “පාසල්” (school) is an adjectival noun which describes the main common noun “ලත්ත” (garden). But according to the Tamil grammar rule, if a noun expresses another noun it cannot be categorized under adjective category. It is classified as noun in Tamil.

Sinhala and Tamil nouns are morphologically inflected based on the case. To indicate case, a suffix is attached. According to Sinhala language rules, it is incorrect to detach these case marking suffixes from the main noun. However, some Sinhala writers tend to separate this case marking suffix from the main noun. So unlike the Tamil language, the Sinhala language has space in between the noun and its case marker. Subsequently, there is a new POS tag added “Case marker” in Sinhala, but not in Tamil. Case marker does not have an English meaning on its own. This tagset has to align with a common noun or proper noun according to the previous tag set alignment in the Sinhala language.

For an example nominative form of ගස - gasa “the tree” can be inflected as ගසට - gasata “to the tree”. ගසට - gasata can be written as ගසට - gasata or ගස ට - gasa ta. In the second case ට - ta has to be tagged as case marker. But in the Tamil language, it will be “மரத்துக்கு” and tagged under the common noun category.

6.1.4 Lexical Divergence

Lexical divergence arises out of the unavailability of an exact translation map for a construction in one language into another language. In Sinhala to Tamil translation, we could come up with three types of lexical divergence.

- The event is lexically realized as main verb in SL but as a different verb in TL. For example,

  පොහොර දමයි

  வலுப்படுத்தல்

  Helps to develop a fight between two people

We notice that in Tamil, the Sinhala phrasal verb 'පොහොර දමයි' is realized by a different verb **வலுப்படுத்தல்** 'develop' which takes an adverbial element පොහොර දමයි 'fertilizing'. The example shows that the divergence pattern not only involves differences in lexical mapping but also in structural mapping between the two languages. Besides, the domain of this type of translation divergence is far from clear. Most of the conflational and inflational as well as some other types of divergences can also overlap with this category. This shows that this category of translation divergence is not well defined in a sense to account for the relevant types of divergence in an exact way.

- Idioms. For example,

  වක්කඩේ හකුරු හැංගුවා වගේ

  தேவை இல்லாததை செய்தல்

  Doing unnecessary thing

In this example වක්කඩේ හකුරු හැංගුවා වගේ is an idiom. Direct meaning is different from indirect meaning. We cannot find exact idiom in Tamil language for this. So here we need to translate into indirect the meaning of that idiom. Here that idiom implies doing unnecessary things. So we have to understand it and need to translate into Tamil proper way.

- Tamil Onomatopoeia (**இரட்டை கிளவி**). For example,

  සීනුව නාද විය.

  மணி கணீர் கணீர் என ஒலித்தது

  The bell rang

Tamil Onomatopoeia refers to the Tamil language words that phonetically imitates, resembles or suggests the source of the sound that it describes. In this example '**கணீர் கணீர்**' is a specific category in Tamil language. We cannot translate into Sinhala language. So we need to eliminate those words when we are translating. When we translate from Sinhala to Tamil, there is a need of adding those words.

There are some other types of divergences also available in the translation between Tamil and Sinhala languages which do not fall under the Dorr's classification. Those are,

- Word order

In Sinhala to Tamil translation, there are some scenarios to reorder one/more words from one place to another place. In that time, we don't reorder the word; it may be grammatical or meaning wise issue. For example,

ලියුම්100

100 **கடிதங்கள்**

100 letters

In this example in Sinhala first ලියුම් occurs and then number 100 occurs. But in Tamil, it is other way round. So we have to have a mechanism to reorder the words.

Another example,

අයි.එම්.ඒ.උදය කුමාර මහතා ,

திரு.ஐ.எம்.ஏ.உதய குமார ,

Mr. I.M.A Uthayakumara,

In this example also Sinhala word 'මහතා' (Mr) need to reorder in front of the name when we translate it into Tamil language.

- Cases for nouns

There are nine cases in Sinhala language as ප්‍රථමා විභක්තිය (praTəma: wibhakthiya) , කර්ම විභක්තිය (karmə wibhakthiya), කර්තෘ විභක්තිය (kathru: wibhakthiya), කරණ විභක්තිය (karaɳə wibhakthiya), සම්ප්‍රදාන විභක්තිය (sampradha:nə wibhakthiya), අවධි විභක්තිය (awadhi wibhakthiya), සම්බන්ධ විභක්තිය (sambandha wibhakthiya), ආධාර විභක්තිය (a:dha:rə wibhakthiya), ආලපන විභක්තිය (a:lapənə wibhakthiya) .

But the usual treatment of Tamil case (Arden 1942) is one where there are seven cases--the nominative (first case), accusative (second case), instrumental (third),

dative (fourth), ablative (fifth), genitive (sixth), and locative (seventh). The vocative is sometimes occupied a place in the case system as an eighth case, even though vocative forms don't participate in usual morphophonemic alternations, nor do they govern the use of any postpositions.

The case markers, the postpositions and the examples in both languages are denoted in table 6.1.

Table 18 The case markers and the postpositions

case	**Sinhala**						**Tamil**			
	animate		**inanimate**				**animate**		**Inanimate**	
	Singular	**Plural**	**Singular**		**Plural**		**Singular**	**plural**	**Singular**	**Plural**
Nominative (praTəma: wibhakthiya)	ගුරුවරයා (teacher)	ගුරුවරු (teachers)	විශ්වවිද්‍යාලය (university)	පාර (road)	විශ්වවිද්‍යාල	පාරවල්	ஆசிரியர்	ஆசிரியர்கள்	பல்கலைக்கழகம்	பல்கலைக்கழகங்கள்
Accusative (karmə wibhakthiya)	ගුරුවරයා (teacher)	ගුරුවරුන් (teachers)	විශ්වවිද්‍යාලය (university)	පාර (road)	විශ්වවිද්‍යාල	පාරවල්	ஆசிரியரை	ஆசிரியர்களை	பல்கலைக்கழகத்தை	பல்கலைக்கழகங்களை
Subjective (kathru: wibhakthiya)	ගුරුවරයා විසින් (by teacher)	ගුරුවරුන් විසින් (by teachers)								
Ablative (karaŋə wibhakt	ගුරුවරයාගෙන් (from	ගුරුවරුන්ගෙන් (from	විශ්වවිද්‍යාලයෙන් (from university	පාරෙන් (from)	විශ්වවිද්‍යාලවලින්	පාරවල්වලින්	ஆசிரியரிடமிருந்து	ஆசிரியர்களிடமிருந்து	பல்கலைக்கழகத்திடமிருந்து	பல்கலைக்கழகங்களிடமிருந்து

hiya)	teacher)	teacher)	)							
Dative (sampradha:nə wibhakthiya)	ගුරුවරයාට (to teacher)	ගුරුවරුන්ට (to teachers)	විශ්වවිද්‍යාලයට (to university)	පාරට (to road)	විශ්වවිද්‍යාලවලට	පාරවල්වලට	ஆசிரியருக்கு	ஆசிரியர்களுக்கு	பல்கலைக்கழகத்துக்கு	பல்கலைக்கழகங்களுக்கு
Instrumental (awadhi wibhakthiya)	ගුරුවරයාගෙන් (from teacher)	ගුරුවරුන්ගෙන් (from teachers)	විශ්වවිද්‍යාලයෙන් (from university)	පාරෙන් (from road)	විශ්වවිද්‍යාලවලින්	පාරවල්වලින්	ஆசிரியரால்	ஆசிரியர்களால்	பல்கலைக்கழகத்தால்	பல்கலைக்கழகங்களால்
Genitive (sambandha wibhakthiya)	ගුරුවරයාගේ (teacher's)	ගුරුවරුන්ගේ (teachers')	විශ්වවිද්‍යාලයේ (university's / of university)	පාරේ (of road)	විශ්වවිද්‍යාලවල	පාරවල්වල	ஆசிரியரின்	ஆசிரியர்களுடைய / ஆசிரியர்களின்	பல்கலைக்கழகத்தின்	பல்கலைக்கழகங்களின்
Locative (a:dha:rə wibhakthiya)	ගුරුවරයා කෙරෙහි (on teacher/ for teacher)	ගුරුවරුන් කෙරෙහි (on teachers/ for teachers)	විශ්වවිද්‍යාලයෙහි (on university / for university)	පාරෙහි (on road)	විශ්වවිද්‍යාලවල /විශ්වවිද්‍යාලෑ	පාරවල්වල/පාරවල්වලෑ	ஆசிரியரிடம்	ஆசிரியர்களிடம்	பல்கலைக்கழகத்திடம்	பல்கலைக்கழகங்களிடம்

Vocative (a:lapən ə wibhakthiya)	ගුරුවරය (teacher !)	ගුරුවරුනි (teachers !)					ஆசிரியரே	ஆசிரியர்களே	பல்கலைக்கழகமே	பல்கலைக்கழகங்களே
Udhdhe:sha wibhakthiya (only in tamil)	ගුරුවරයා ළඟ	ගුරුවරුන් ළඟ	විශ්වවිද්‍යාලය ළඟ	පාර ළඟ	විශ්වවිද්‍යාල ළඟ	පාරවල් ළඟ	ஆசிரியரிடம்	ஆசிரியர்களிடம்	பல்கலைக்கழகத்திடம்	பல்கலைக்கழகங்களிடம்
Saha:rtha wibhakthiya (only in tamil)	ගුරුවරයා සමඟ	ගුරුවරුන් සමඟ	විශ්වවිද්‍යාලය සමඟ	පාර සමඟ	විශ්වවිද්‍යාල සමඟ	පාරවල් සමඟ	ஆசிரியருடன்	ஆசிரியர்களோடு	பல்கலைக்கழத்தினோடு	பல்கலைக்கழகங்களோடு

Definiteness		**indefiniteness**	
Animate	Inanimate	Animate	Inanimate
ගුරුවරයා (ගුරුවර + ආ) (insertion of ය් and sandhi among inserted character and suffix)	විශ්වවිද්‍යාලය (විශ්වවිද්‍යාල+අ) (insertion of ය් and sandhi among inserted character and suffix)	ගුරුවරයෙක් (ගුරුවර +එක්) (insertion of ය් and sandhi among inserted character and suffix)	විශ්වවිද්‍යාලයක් (විශ්වවිද්‍යාල+අක්) (insertion of ය් and sandhi among inserted character and suffix)

- Tenses for verbs

There are three tenses such as Past tense, Present tense and Future tense basically, while Sinhala language has two tenses such as Non-past tense and Past tense. Although the tenses were categorized into three except other conditional tenses (Dilshani,Dias:2017) as past, present and future in Sinhala language in early grammar, there are only two tenses are available by combining present and future as one in Sinhala language as past and non-past as the same conjugational forms can be used for both present and future tenses. Although, in Tamil language consists of three separate tenses as past, present and future tenses. Additionally, there are separate conjugational patterns for three tenses as Table 6.2.

Table 19 Tenses and Examples

Tense		Sinhala	Tamil
Past tense	I studied	මම ඉගෙන ගත්තෙමි	**நான் படித்தேன்**
Present tense	I am studying/ I study	මම ඉගෙන ගනිමි/ ගන්නෙමි	**நான் படிக்கிறேன்**
Future tense	I will study	මම ඉගෙන ගනිමි/ ගන්නෙමි	**நான் படிப்பேன்**

- Determiner System

There are four types of determiners to specify here and there. Those are අරගේ, මෙහෙ, ඔහේ and එහේ. But Tamil has only two determiners such as **அங்கே** and **இங்கே**. The divergence of the determiner system of Sinhala and Tamil languages are given in table 6.3.

Table 20 Divergence of the determiner system of Sinhala and Tamil

	Sinhala	Tamil
PERSON	එයා, ඒකා, ඒකි, ඌ, එතුමා, උන්නාන්සේ (he/she) එයාලා, ඒකලා, ඒකිලා, උන්, එතුමන්ලා, උන්නාන්සේලා	**அவர் அவள்** **அவர்கள்**

	(they) ඔයා, උඹට, තෝ, තමුසේ, ඔහේ, ඔබතුමා, තමුන්නාන්සේ (you) ඔයාලා, උඹලා, තොපි, තමුසෙලා, ඔහේලා, ඔබතුමාලා (you)	நீ நீங்கள் நீங்கள்
PLACE	ඔහේ (close to the hearer), අරහේ (far from both but in the vicinity), එහේ (there), මෙහේ (here)	அங்கே இங்கே
TIME		
INANIMATE	ඒක (that), ඕක, මේක(this), ඒවා (those), මේවා (these)	அது , இது , அதுகள் , இதுகள்

- Passive voice sentence

In Sinhala language Passive voice is mentioned by a particular word 'විසින්', but Tamil doesn't have a particular word to specify passive voice sentences. So when we translate from Sinhala to Tamil, we have to eliminate that word.

By studying the divergence between those languages, we have come up with aligning POS tagsets within these two languages and how to come overcome those issues by some pre-processing and post-processing techniques in statistical machine translation. Those results are given in next sections.

6.2 Semi-automatic alignment between Tamil and Sinhala POS tagsets

Through the experiment, there are some possible relationships holding between BIS tagset and UOM tagset. In this section, the details of four types of relationship and the examples are focused. The results of POS tagset alignment of Tamil and Sinhala languages after manually proven are tabulated in Table 6.4. Results are based on word alignments and two linguists' opinion. There are 8 equal relationships, 22 subsumption relationships, 1 complex relationship and no non mapped relationships.

Table 21. Alignment of BIS tagset and UOM tagset

UOM Tags	**BIS Tags**	**Example**		
Common Noun	Common Noun/Echo words	மரம்	ගස	Tree
Adjectival Noun		பாடசாலை,	පාසල්	School
Case marker	Common/proper	க்கு ,உடைய	ට, ගේ	to,’s
Proper noun	Proper noun	ஜான்	ජොන්	John
Pronoun/Deterministic Pronoun	Personal Pronoun	நான், நீ	මම, ඔබ	I, you
Pronoun	Reflexive Pronoun	தான்	-	Myself
	Reciprocal Pronoun	ஒருவருக்கொருவர,அவனவன்	එක එක්කෙනාට,ඔවුනොවුන්	each other

Questioning Pronouns	Question words	என்ன, எப்படி	කුමක්ද, කෙසේද	what, how
Question-Based Pronouns	Relative Pronoun	எங்கே, எது	කොහේ, කවර	where, which
Determiners	Deictic	இவன், இவள்	මේ, සියලු	this, all
	Relative	அவ்வீடு, இவ்வீடு	ඒ ගෙදර, මේ ගෙදර	That home, this home
Verbal Participle	Verbal participle	பார்த்து	බලා	Looked
Verb finite	Verb finite	செய்தான்	කළේය	Did (he)
Preposition in compound verb		-	ඉටු, සිදු	-
Nouns in Compound Verb		படிக்கின்றான்	පාඩම් කරනවා	Study
Adjective in Compound Verbs		கூட்டபடுகின்றது	වැඩි කරනවා	Increasing
Nipathana		போதும், காணாது	ඇති, නැති	Enough/ not having
Modal auxiliary	Verb auxiliary	முடியும், வேண்டும்	හැකි, යුතු	Can, should

Verb Non-Finite	Infinite Verb	விழ	වැටීමට වගේ	like to fall
	Conditional Verb	நடந்தால்	ඇවිද්දොත්	If walk
Verbal Noun	Verbal Gerund	படித்தல்	ඉගෙනීම	Studying
	Verbal noun	படிப்பு	-	Study
Adverb	Adverb	விரைவாக	වේගයෙන්	Fast
Adjective	Adjective	மிருதுவாக	සුමුදු	Smooth
	Relative Participle	நடந்த	ඇවිද	Walked (kid)
Conjunction	Coordinator	உம், மற்றும்	හෝ, සහ	Or, and
	Subordinator	என்று, என	යනු, යැයි	That
Particle	Default Particles	மட்டும், கூட	ය, ද, ම	Only, also
	Classifier	அட்டும்	-	-
	Intensifier	அதி, வேக, மிக	ඉතා	Most, speed
	Negation	இல்லை	නෑ, නැත	No

Interjection	Interjection	ஐயோ	අයියෝ	Oh
Postposition	Postposition	பற்றி,குறித்து	ගැන	Related
Number	Cardinal	ஒன்று, 1	එක, 1	One, 1
	Ordinal	முதல், இரண்டாம்	පළමුවන, දෙවන	First, second
Punctuation/Full stop	Punctuation	/?,:''	/?,:''	/?,:''
	Symbol	$, &,*,(	$, &,*,(	$, &,*,(
Foreign word	Foreign Residuals	கார்	කාරය	Car
Abbreviation	Unknown	மு.ப	පෙ.ව	a.m

6.2.1 Equal relationship

There are some POS alignments which hold an equal relationship. Equal relationship implies one language tagset can equally align with another language tagset. As mentioned in Table 1, some POS alignments fall under the equal relationship. The adverb in the Tamil language can be directly mapped to Sinhala language adverb node. Modal auxiliary in UOM tagset and Verbal auxiliary in BIS tagset are equally aligned. Verbal participle, Common noun, Postpositions, Foreign words and Punctuation in both languages are fallen in the equal relationship as it has same features. Questioning pronouns words are used to ask a question. So that is equivalently aligned with question words in BIS tag set.

6.2.2 Subsumption relationship

In most of the cases, a POS tag in the Sinhala language is not mapped directly to Tamil language POS tag. Most of those tags fall under subsumption relationship.

Nipathana is a category in the Sinhala language, but which does not have direct mapping tag in the Tamil language. So Nipathana does have to map with the finite verb category in the Tamil language (subsumption ⊆, ⊇). Conjunction is specialized into subordinator and coordinator in the Tamil language. So these two subcategories are aligned to parent node conjunction in Sinhala language (subsumption ⊆ Relationship). This mostly occurs when a number of aspects used in the specialization of a POS tag differ between languages. BIS tagset does have five categories of pronouns while there are only four categories in UOM tag set. As a result, we are not able to equal align those tags. The Personal, Reflexive and Reciprocal pronouns from BIS tagset are subsumptionly aligned with Pronoun tag in UOM tag set. Deterministic pronouns in UOM tagset are aligned to personal pronouns in BIS tag set. Furthermore, the category of personal pronouns can contain other words except for deterministic pronouns. Question-based pronouns are used to show the uncertainty of a noun/noun phrase of interest. So this tag aligns with the Relative pronoun in BIS tag set. But Relative pronoun can contain other words than question-based pronouns.

E.g: I don't know who did this.

இதை யார் செய்தது என்று எனக்கு தெரியாது.

මෙය කළේ කවුදැයි මම නොදනිමි.

There are two types of demonstrative in BIS tag set while UOM tag sets have only one category. The subcategories Deictic and Relative are aligned to Determiners tag. Particles are further divided into five sub-categories in BIS tag set while there are only a parent node Particles in UOM tag set. Hence, the subcategories are mapped to Particles in UOM tagset using subsumption relationship. General, ordinal and cardinal are the three categories of Quantifiers in BIS tag set. Yet, UOM tag set only have Number category. Thus, three subcategories are aligned with Number category. Full stop in UOM tagset does have subsumption relationship with punctuation in BIS tag set. Like that, Symbol in BIS tag is aligned with punctuation category of UOM tag set. As BIS tagset do not have a proper tag for Abbreviation in UOM tagset, it takes the subsumption relationship with Unknown tag. Echo words in BIS tag set are aligned to the Common noun in UOM tag set.

A noun in Compound Verb is another category of the noun in the Sinhala language. It is a combination of noun and verb. The noun which makes compound verb is called as nouns in the compound verb. There is no matching translation in English and Tamil since all compound verbs in the Sinhala language is a normal verb in English and Tamil. In this example, First part of the verb is identified as 'Noun in the compound verb'. So this 'Noun in Compound verb' tag is subsumptionly mapped with Finite verb tag of the BIS tagset.

E.g. එයා පාඩම් කරනවා.
He is studying.
அவன் படிக்கிறான்.

The adjectival noun is a common noun that acts as an adjective to describe another noun. When a common noun is used as an adjectival noun, it always takes the base, plural form of the common noun. For example, in a noun phrase like 'පාසල් වත්ත (school garden)', 'පාසල් (school)' is an adjectival noun which describes the main common noun 'වත්ත (garden)'. But according to the Tamil grammar rule, if a noun expresses another noun it cannot be categorized under adjective category. So those 'Adjectival noun' is mapped with common noun in BIS tagset.

Further, adjectives are categorized into three subcategories Adjective, Adjectival Noun, and Adjective in Compound Verbs. As we saw above, Adjectival Noun tag is aligned to Common noun tag. The adjective in Compound Verb is a combination of Adjective + Verb. The first word in such compound verbs will be tagged as Adjective in compound verbs. In the example 'වැඩි කරනවා (increase)', වැඩි is an adjective and කරනවා is a verb. But Tamil we can write this as 'கூட்டபடுகின்றது'. Hence, there is no matching translation in Tamil for the adjective in the compound verb, since all compound verbs in Sinhala is a normal verb in Tamil. Thus 'Adjective in the Compound verb' is mapped with Finite verb tag of the BIS tagset. Remaining subcategory 'Adjective' is aligned to Adjective in BIS tag set.

Non-finite and finite verb forms often constitute mixed categories from the syntactic point of view. The syntactic properties of participles overlap with adjectives. Relative participle from verb category in BIS tagset also map with adjective in UOM tag set. Similarly, gerunds and verbal nouns BIS tagset is aligned to Verbal noun in UOM tagset. At the same time, however, they retain their verbal arguments. Usually, these

words are tagged as forms of verbs. Likewise, infinite verb and conditional verb in BIS tag set are aligned to non-finite verb category in UOM tag set.

Some other categories in UOM tagset also fall under the Verb category of BIS tagset. Similar to 'Adjective in Compound Verb', 'Preposition in the compound verb' is one of the categories in the UOM tagset which does not have a meaning by them but, when combined with another verb, make up a compound verb. In the example 'ඉටු කරයි (does)', ඉටු is a preposition and කරයි is a verb. But Tamil we can write this as 'செய்கிறார்'. Hence, there is no matching translation in Tamil for the preposition in the compound verb, since all compound verbs in Sinhala is a normal verb in Tamil. Thus 'Preposition in the Compound verb' is mapped with Finite verb tag of the BIS tagset.

Nipathana is a tag in UOM tagset which is used alone in some contexts and as a postposition. But Tamil language does not have an exact match for this category. This category is mapped with Finite verb tag by considering the usability of this category.

E.g ඇති (Enough) - **போதும்**,
නැති (not having) – **கிடையாது**

6.2.3 Complex relationship

Some features in POS tagset are unique to the particular language. Those features may map to another category or categories when we come to alignment. There are some complex alignments when we try map POS tagsets of Sinhala and Tamil language. Hence, we went deep in the grammar of both languages to find out the relationship for those categories.

Sinhala and Tamil nouns are morphologically inflected based on the case. To indicate case, a suffix is attached. According to Sinhala language rules, it is incorrect to detach these case marking suffixes from the main noun. However, some Sinhala writers tend to separate this case marking suffix from the main noun. So unlike the Tamil language, the Sinhala language has space in between the noun and its case marker. Subsequently, there is a new POS tag added "Case marker" in Sinhala, but not in Tamil. Case marker does not have an English meaning on its own. This tag set has to align with a common noun or proper noun according to the previous tag set alignment in the Sinhala language. So this alignment falls into the composite relationship. For an example nominative form of ගස - gasa "the tree" can be inflected as ගසට - gasata "to

the tree". ගසට - gasata can be written as ගසට - gasata or ගස ට - gasa ta. In the second case ට - ta has to be tagged as case marker. But in the Tamil language, it will be "மரத்துக்கு" and tagged under the common noun category. This correspondence is fallen into compa osite relationship. POS alignment depicts the grammar of the language to a certain level.

6.3 Hierarchical phrase-based model machine translation system

The evaluation scores of the aforementioned three language pairs in both the directions and the sample translations from the developed HPM SMT are described in this section. In each language pair, we trained the SMT with and without HPM and evaluated its translation quality by measuring the BLEU score of the translation of test data set. Even though these language resources are sparse, we have achieved much better BLEU score for the entire set of language pairs. The scores of six different experimental setups are tabulated in Table 7.5. A comparison of the developed hierarchical phrase-based translation system with the traditional phrase-based system was also carried out for the same dataset.

It can be noted from Table 6.5, that the hierarchical phrase-based model system got better BLEU scores compared to the traditional Phrase-based model approach for Tamil to English, English to Tamil, Malayalam to English and English to Malayalam. While those differences are less since the dataset size is small, the percentage of the difference is high. These results show that usefulness of hierarchical phrase-based model is significant when it is different in sentence structure between the languages getting translated. Nevertheless, for the translation of Tamil to Sinhala and Sinhala to Tamil, it could be noticed from Table 7.5 that the traditional phrase-based model system got better BLEU scores compared to the hierarchical phrase-based model approach. The main reason behind this is that both Tamil and Sinhala language share same sentence structure and morphologically rich. Further, the Tamil-Sinhala corpus is the smallest among the three which causes sparseness in training data. HPM is sensitive to sparse data and that could have further reduced the translation quality in this case. These observations show that the HPM is most useful in language pairs varied by sentence structure but would affect the quality of the translation if the languages share the same sentence structure.

Table 22 Comparison of BLEU evaluation score with traditional Phrase-based model

	BLEU Score		Differentiation (%)
	Traditional Model	**Hierarchical Model**	
Tamil to English	3.16	3.42	8.23
English to Tamil	1.17	1.73	47.863
Malayalam to English	4.22	4.40	4.26
English to Malayalam	2.88	3.310	14.93
Tamil to Sinhala	*****14.88	11.18	(-20.16)
Sinhala to Tamil	*****13.61	10.73	(-21.17)

The results also show that BLEU score increase is higher from English to Tamil or Malayalam compared to the other direction. As we know Tamil & Malayalam are morphologically richer than English. In these cases, HPM leverages the morphological divergence between these languages in its favor. Also from the results, it can be noted that the translation from morphologically rich languages (Tamil, Malayalam) to morphologically poor languages (English) gives better BLEU score in traditional SMT and HPM SMT compared to another way around. Even though Sinhala is a morphologically rich language, the translation from Tamil to Sinhala shows higher results as Tamil language is morphologically richer than the Sinhala language. These observations show that the translations from morphologically rich languages to morphologically poor languages give better result compare to other direction.

Also, English to Tamil got the highest percentage of increase in BLEU score due to HPM compared to traditional SMT (47%), and Sinhala to Tamil got the highest decrease in BLEU score percentage (21%). From the results, it can be observed that the translations from morphologically poor languages (English) to morphologically rich languages (Tamil, Malayalam) give more improvement using HPM model. So, the usefulness of HPM is significant when the divergence of morphology and divergence of sentence structure.

Figure 7.1 shows how the decoder performs translations of the test dataset using the chart decoder for hierarchical phrase-based model. For the input Tamil sentence "ஆரம்பத்திலே சிறிய உடற்பயிற்சி செய்யுங்கள்.", the sentence is translated as "Start with light exercise".

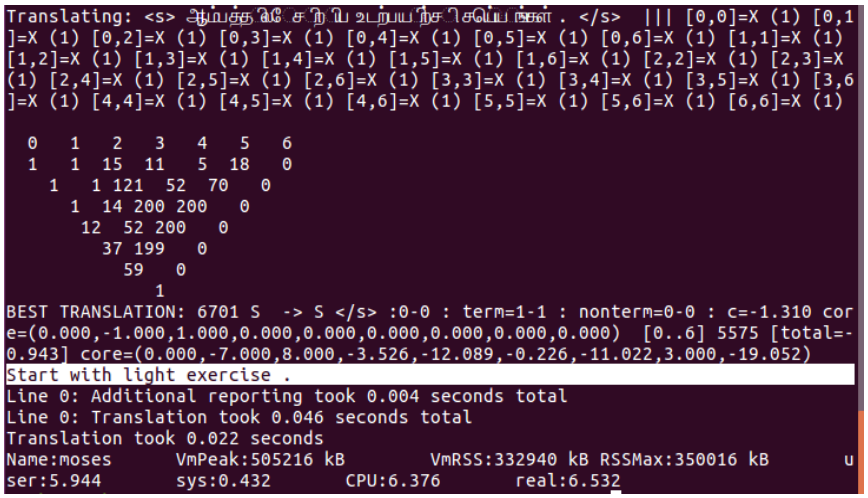


Figure 6.18 Working of Hierarchical Phrase based decoder for Tamil to English translation

Some examples of translation generated by translation system developed in this study are provided in Table 6.6. Two examples of each different translation have been listed here. Some of the examples are not perfect translations. This may occur since the South Asian languages are rich in morphology compared to English, there may be noise in training data, out of vocabulary, misordering of words, wrong alignment of phrases, inappropriate translation to the context and harder sparse-data problems due to vocabulary that combines words from various sources. However, there are some examples below which show hierarchical phrase-based model helps to reorder the sentences.

Table 23 Some examples of translation generated by the translation system developed in this study

	Input	**Output**
Tamil to English	சைட்டிகா மற்றும் சிலிப்புடிஸ்க் நோயாளி இந்த பயிற்சி செய்யாதீர்கள்	The patients of sciatica and slip disk should avoid its practice
	பூங்காவிற்கு பைக் எடுத்துச்செல்ல அனுமதியில்லை.	The a rate of a take her அனுமதியில்லை
English to Tamil	Drink plenty of water	நன்றாக தண்ணீர் குடியுங்கள்
	Chew the sugar-free chewing gum	சர்க்கரை இல்லாத சூயிங்கம் மெல்லவேண்டும்
Malayalam to English	പതിവായിട്ട് ദന്തനിരീക്ഷണം ചെയ്യണം .	Get the teeth checked-up regularly
	നെഓറാ താഴ്വര ദേശീയ ഉദ്യാനത്തിൽ അനേകം ഓർക്കിഡുകളുണ്ട് .	Neora Valley National Park is Approximately 150 species of

		orchids are found
English to Malayalam	New Digha is the new tourist spot of Digha	ചിൻഡ്യിയിലെ കമരൂനാഗ് . ഗിവഗുഫ തുടങ്ങിയവ കാണേണ്ടതാണ് .
	There is ` Forest Hut ' in Sanarali 3 kms ahead	3 കി.മീ. സനാരലിയിൽ ഫാം റെസ്റ്റ് ഹട്ട് ഉണ്ട് .
Tamil to Sinhala	கலந்துரையாடல் நிகழ்வு செயலாளரின் தலைமையில் இடம்பெற்றது	සංවාද සිද්ධිය ලේකම්ගේ ප්රධානත්වයෙන් පැවැත්විණ .
	பணிக்கான கணக்குப் பிரிவு	කාර්යය ගිණුම් අංශය
Sinhala to Tamil	ජල පොම්ප අළුත්වැඩියා අංශය	நீர் திருத்துதல் பிரிவு
	ජල සැපයුම් අංශය	நீர் வழங்கல் பிரிவு

6.4 POS Integration to SMT system

The evaluation scores of the Sinhala to Tamil translation and the sample translations from the developed POS integrated SMT are described in this section. We have trained the SMT without a factored model and five different types of factored model and evaluated its translation quality by measuring the BLEU score of the translation of test data set. Even though these language resources are sparse, we have achieved much better BLEU score for the Sinhala to Tamil translation.

All the developed models are evaluated with the same test-set which contains 300 Sinhala sentences. The well-known Machine Translation metrics BLEU [11] and human evaluation are used to evaluate the developed models. In addition to that the existing “Google Translate” online Sinhala-Tamil machine translation system is also evaluated to compare with the developed models. The results are in terms of BLEU score and it is shown in Table 6.7. In figure 7.2, X-axis represents the various machine translation models and Y-axis denotes the BLEU scores.

From the graphs in the figures, even though it is shown that the proposed system (SoPOS+TaPOS+factLM) improves the BLEU score slightly compare to other developed models and “Google Translate” system, the manual evaluation shows better result by integrating POS into SMT. Human evaluation details are shown in below subsection. In this output, both sentences are failed to produce a good flow of the sentence. The grammatically correct output is not available in alternate translations also.

Table 24 The results are in terms of BLEU score for Baseline and POS integrated models

Models	BLEU Score
Baseline (BL)	**29.8**
Factored system with adding POS tag to the source side + normal LM (SoPOS)	**28.22**
Factored system with adding POS tag to the target side + normal LM (TaPOS)	**29.07**
Factored system with adding POS tag to the source side + POS tag to the target side + normal LM (SoPOS+TaPOS)	**30.33**
Factored system with adding POS tag to the source side + POS tag to the target side + factored LM (SoPOS+TaPOS+factLM)	**30.54**
Google Translate	**7.86**

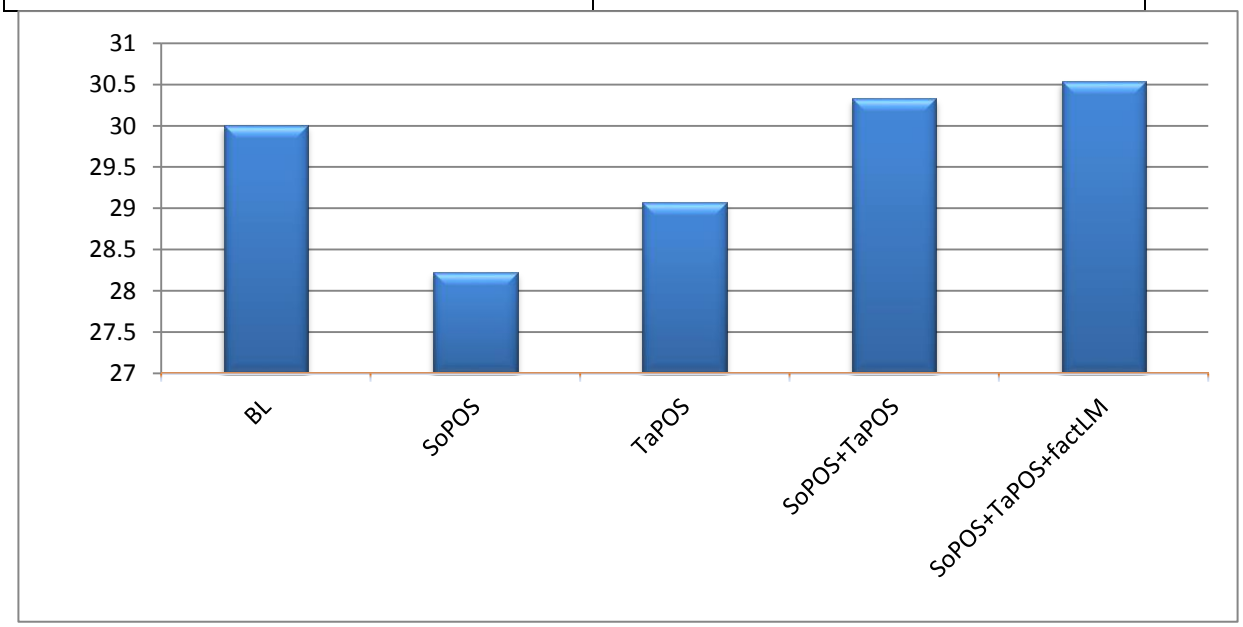


Figure 6.19 Graph of various machine translation models and the BLEU score

6.4.1 Human Evaluation

Human evaluation is a noticeable method for evaluating machine translation output. Here evaluators look at the output and judge by hand whether it is correct or not. Bilingual evaluators who understand both the source and target language are best qualified to make this judgment. Such bilingual evaluators are not always available, so we often have to resort to monolingual evaluators who understand only the target language but are able to judge system output according to reference translation. A more common approach is to use a graded scale when causing judgments from the human evaluators. Much more common to have human evaluators simply assign a scale directly using fluency/adequacy scales. Moreover, correctness may be too broad a measure. It is, therefore, more common to use the two criteria such as fluency and adequacy.

Fluency: Is the output good fluent in the target language? This involves both grammatical fluency correctness and idiomatic word choices.

Adequacy: Does the output convey the same meaning as the input sentence? Is part of the message lost, added, or distorted?

By considering those factors we came up with 4 point-scale system. The scale points and description of the scales are given in table 6.8.

Table 25 4 point scale system for human evaluation

Rating	The flow of the target language sentence	Meaningful Translation of source language
4	Flawless target language and contains all information in the reference translation	No not translated words
3	Good Tamil and contains most of the information in the reference translation	At most 2 words are not translated
2	Non-native Tamil, without not the proper flow of the sentence and contains reasonable information in the reference translation	More than 2 words are not translated

1	Incomprehensible Tamil and contains very little information in the reference translation	A lot of not translated words

In this research three humans who are good in Tamil language are used to evaluate the system. Human evaluation is done in four steps. First, we have compared POS integrated system (SoPOS+TaPOS+factLM) with traditional SMT (BL) and find out how many sentences are same between them and how many sentences have differed in translation when we using SoPOS+TaPOS+factLM and BL. The details are shown in the below pie chart 6.3.

Same translations in both systems: 140

Different translations between both systems: 160

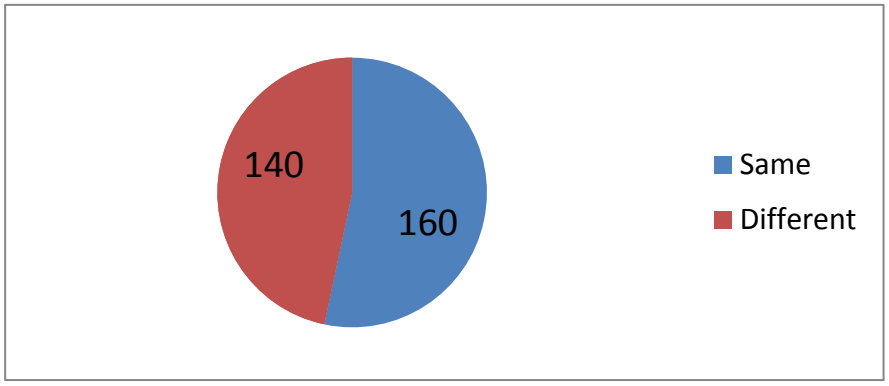


Figure 6.20 Pie chart for the sentences which are same and different between POS integrated model and Baseline

Then the sentences which are same in both systems are compared with the human reference. 4 point scale system mentioned in table 6.4 is used to evaluate the translations. The detail of the evaluation is given in Figure 6.4.

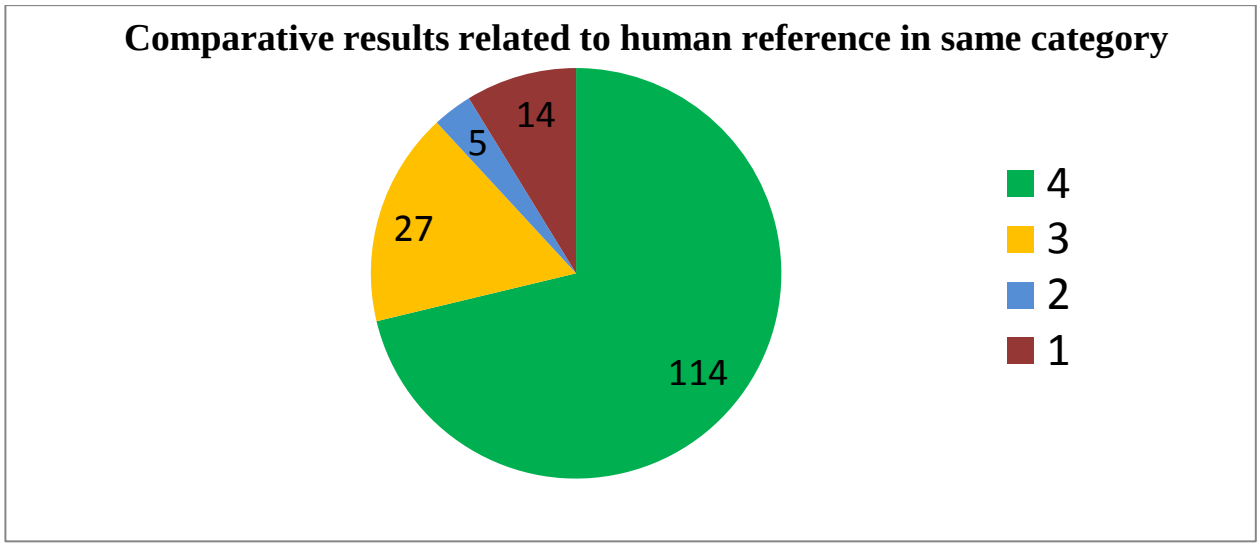


Figure 6.21 Pie chart for Comparative results related to human reference in the same category

After that, the translations which are different between POS integrated system and Baseline systems are compared with each other. Here we have checked the translations within each other and found out which translation is better comparable to another one. The result of the evaluation is given in Table 6.9. From the table we could see 140 different sentences, 96 translations are better in POS integrated model to compare to the baseline model. 44 translations are better in baseline model to compare to POS integrated model. But it is not sure that the better translations are good enough like human reference.

Table 26 Results of the comparison between the translations which are different between POS integrated system and Baseline systems

	POS Integrated	**Traditional**
Better results	**96**	**44**

As we are not sure the above better translations by comparing with each other are good enough like human reference, we have done evaluations to compare the better translations with human reference. So 96 better translations belong to POS integrated model and 44 better translations belong to baseline model are evaluated with human reference using 4 point scale system mentioned in table 6.9. Pie chart for

comparative results of better translations belong to POS integrated model and baseline model related to human reference is shown in 6.5 and 6.6 respectively.

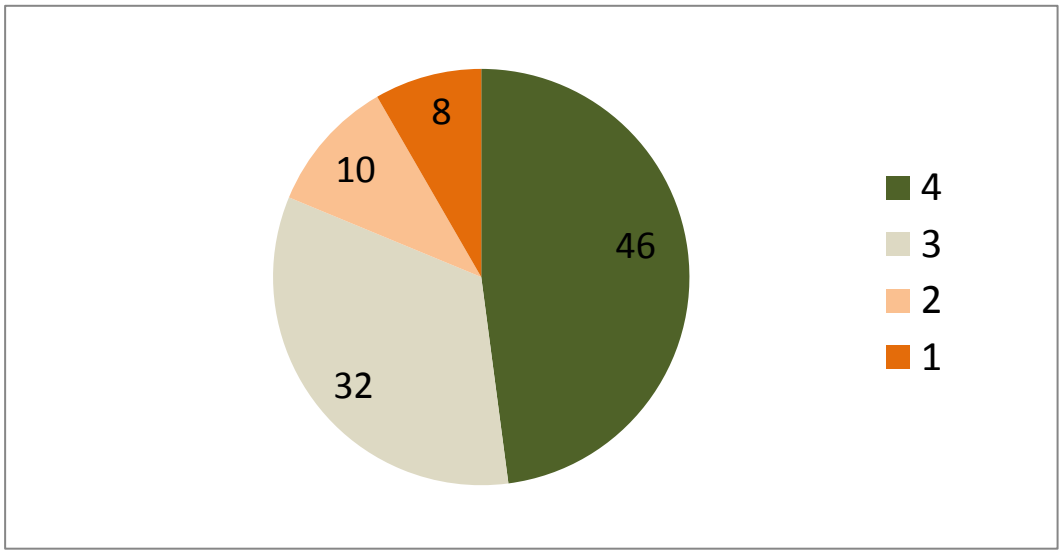


Figure 6.22 Pie chart for Comparative results of better translations belong to POS

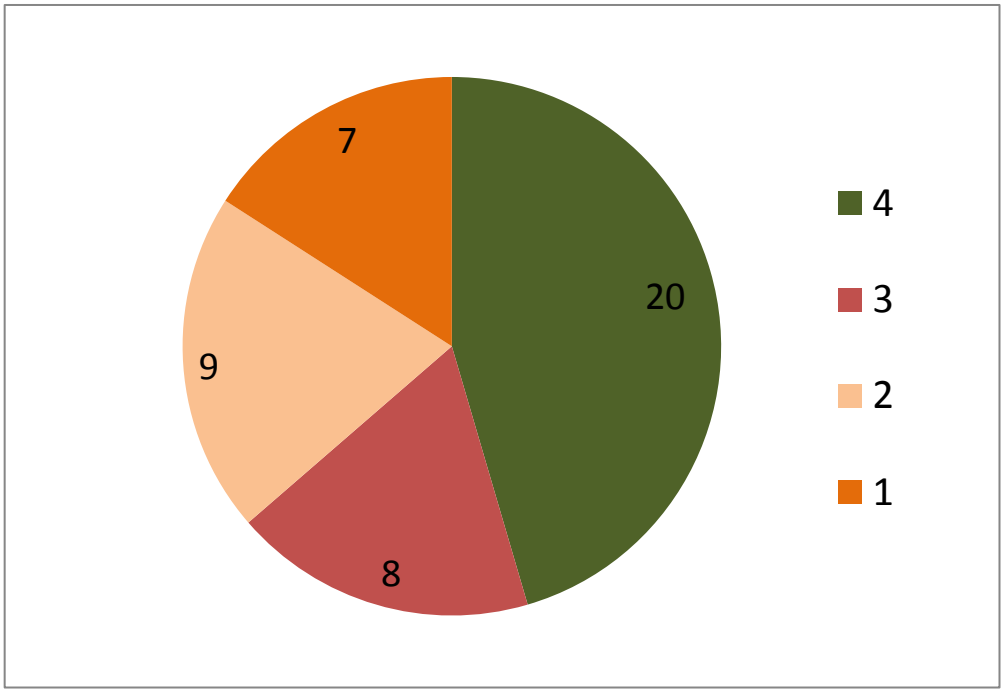


Figure 6.23 Pie chart for Comparative results of better translations belong to baseline model related to human reference

From the experiment of POS integrate model and baseline model, we could found out some observations. Those observations and analysis study of those observations is listed below.

- **Word reordering**

With POS	Without POS
100 கடிதங்கள்	கடிதங்கள் 100
திரு . ஆர் . எம் . வேரஹர	ஆர் . எம் . வேரஹர

In this example ‘මහතා’ is translated as ‘திரு’ and in the proper position in the POS integrated model. But in baseline system even the translation didn’t happen. ‘මහතා’ can be translated either ‘அவர்கள்’ or ‘திரு’. As most of the sentences in parallel corpus contain ‘මහතා’ ‘அவர்கள்’ combination, baseline system tends to translate ‘මහතා’ to ‘அவர்கள்’ or not translating. But in POS integrated system, the sentences are tagged using POS before giving to the system. In that case, according to the position, ‘මහතා’ which is translated as ‘அவர்கள்’ is tagged as proper noun and ‘a මහතා’ which is translated as ‘திரு’ is tagged as verb infinite. So according to the position of that word, it is translating correctly.

Reordering in traditional phrase models is typically modeled by a distance-based reordering cost that discourages reordering in general. Reordering is often limited to movement over a maximum number of words. The lexicalized reordering model learns different reordering behavior for each specific phrase pair. So it can only handle local reordering which permits moves within a window of a few words. Like above example, when the name is long, baseline system unable to reorder the word ‘திரு’ in front of the name. Additional information such as part-of-speech may be helpful in making reordering. In factored model, POS alignment table also creates along with word alignment table. Through POS alignment table reordering patterns can also be learned over order of the part-of-speech tags in the sentences. That’s why we could observe reordered sentences in POS integrated system.

- **Context-aware**
    - ‘වී’ in වී ඇත
        - Noun : நெல் (Paddy)

- Verb :செயற்படுத்தப்பட்டுள்ளது (has happened)

In this example 'වී' can be translated as 'நெல்' or 'செயற்படுத்தப்பட்டுள்ளது' according to the noun or verb. In baseline system, there is no way to specify whether it is noun or verb. So according to the no of occurrences in the parallel corpora, it will select one among the all possible translations. But in POS integrated system, we are able to give the tag whether it is noun or verb. So according to the tag, it is translating correctly.

- **Better word choice**
    - **தொலைக்காட்சி** vs **ரூபவாஹினி** in the sentence "ශ්‍රී ලංකාව තුළ ග්‍රාහකයින්ගෙන් මුදල් ලබා ගැනීමේ පදනම මත (Pay TV) රූපවාහිනී සේවාවන් පවත්වාගෙන යාම"

In this example in some scenarios රූපවාහිනී is tagged as proper noun and common noun. When it is common noun it is translating as **தொலைக்காட்சி** and when it is proper noun, it is translating as **ரூபவாஹினி**. So here it is learning pattern according to the context and tag it is translating the source sentence to target sentence.

- **Translating conjunction words**

In baseline system conjunction words and main words are considered as the same word when we are tokenizing. But in the POS integrated system when we are tagging the sentences, conjunction words also tagged as conjunctions. So in the translation, each conjunction words also translated properly as they are considered as another word.

- **Transliteration**
    - 'Diploma in' is transliterated as "Dஇப்லோம இந்"

In the Sinhala side parallel corpus, when we have initials of the name in English, it is transliterated directly to Tamil. So for some English alphabets, there are transliterated Tamil characters in the parallel corpus. In baseline system, those initials and names are considered as a phrase. So baseline system is unable to find out the transliterated character of the English alphabet. But in POS integrated system, when we tagging,

each alphabet are considered as a different token. So when we give new sentence with some English alphabets, it is able to find out the transliterated version of that alphabet from the training set.

- **Abbreviations**
    - வ . செ . தி . கா . அ is an abbreviation which translated correctly in POS integrated system

Like above scenario, in baseline system, those the characters in the abbreviation altogether consider as a phrase. So baseline system is unable to find out the translated character of those characters. But in POS integrated system, when we tagging, each character are considered as a different token. So when we give new sentence with some abbreviation which is not directly in the training data, but each character of the abbreviation and translation of that characters are in the parallel corpora, it is able to find out the transliterated version of that alphabet from the training set.

6.5 Preprocessing based on chunking

The evaluation scores of the Sinhala to Tamil translation and the sample translations from the developed SMT with pre-processing based on chunking are described in this section. The evaluation scores for the preprocessing techniques based on chunking such as finding collocation words from PMI, NER based chunking, and POS based chunking are discussed in individual subsections. We have trained the SMT without the pre-processing technique to compare the translation quality. The evaluation of translation quality is done by measuring the BLEU score of the translation of test data set. Even though these language resources are sparse, we have achieved much better BLEU score for the Sinhala to Tamil translation.

6.5.1 Results for PMI based chunking

All the pre-processed models using PMI based chunking are evaluated with the same test-set which contains 300 Sinhala sentences. The well-known Machine Translation metrics BLEU is used to evaluate the system. Here results of selected top 100 to 1000 collocated words when the frequency is 5, based on the PMI score in both languages and top 500 and 1000 collocated words when the frequency is 10, are discussed here. We selected only 500 and 1000 collocated words when the frequency is 10 as they didn’t give a good result like frequency is 5. The results are in terms of BLEU score

and it is shown in Table 6.10. In figure 6.7, X-axis represents the various different size of chunked words based on PMI score and Y-axis denotes the BLEU scores.

Table 27 The results are in terms of BLEU score for all models

Models	BLEU score
Baseline	29.8
Top 100 and Frequency=5	29.63
Top 200 and Frequency=5	29.14
Top 300 and Frequency=5	30.41
Top 400 and Frequency=5	30.44
Top 500 and Frequency=5	**33.36**
Top 600 and Frequency=5	29.81
Top 700 and Frequency=5	28.79
Top 800 and Frequency=5	30.03
Top 900 and Frequency=5	29.71
Top 1000 and Frequency=5	28.37
Top 500 and Frequency=10	28.3
Top 1000 and Frequency=10	25.90

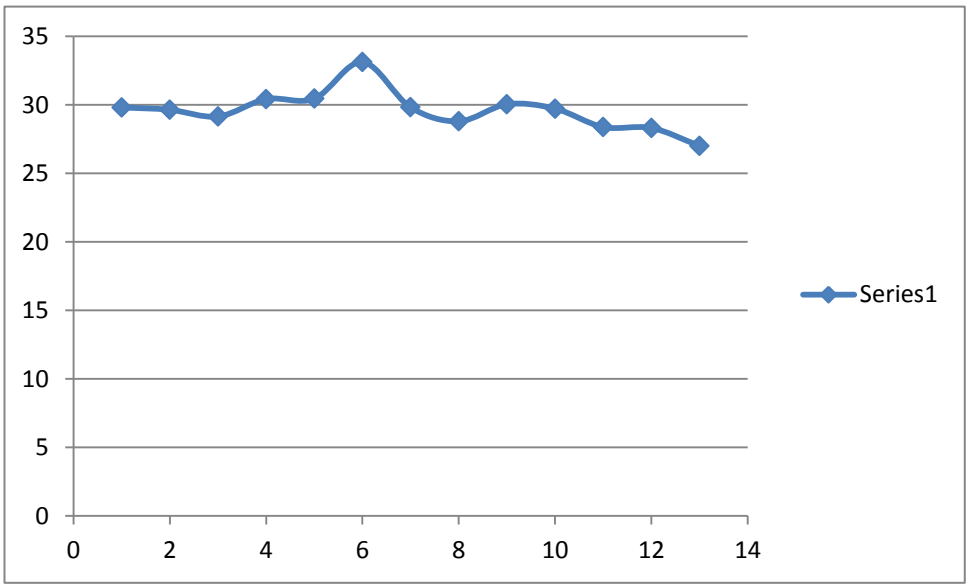


Figure 6.24 Graph of various different sizes of chunked words based on PMI score and the BLEU score

From the graphs in figure 6.7, it is clearly shown that the proposed pre-processing method improves the BLEU score compare to the baseline system. When select top 300, 400,500, 600 and 800 when the frequency is 5, we could observe better translation

quality compare to the baseline system. But within that when we select top 500, we could observe drastic increase.

Within 300 sentences, we could observe improvements in 88 sentences using pre-processing techniques in translation compared to the baseline system. 178 sentences have same translations in both models. And we could observe 34 impacted translations compare to the baseline system. Some sample translations generated by both systems are given in Table 6.11.

Table 28 Sample translations of both models

Input	Output (pre-processed)	Output(Baseline)
ප්‍රමුඛතා කාර්යයක් සේ සලකා මේ සම්බන්ධයෙන් ඔබ ලබා දෙන සහයෝගය බෙහෙවින් අගය කරමි .	முன்னுரிமை அடிப்படையிலும் செயல் கருதி இது தொடர்பாக நீங்கள் வழங்கும் ஆதரவை மிகவும் பாராட்டுகின்றேன் .	இருந்தும் செயல் ஒன்றாகக் கருதி இது தொடர்பாக நீங்கள் வழங்கும் ஆதரவை மிகவும் பாராட்டுகின்றேன் .
අදාල චෙක්පත ලැබුණ බව අමාත්‍යාංශය වෙත නොපමාව දැනුම් දෙන මෙන් කාරුණිකව දන්වා සිටිමි .	உரிய காசோலையை பெற்றன என அமைச்சுக்கு தாமதமின்றி அறியத் தருமாறு தயவுடன் அறியத் தருகிறேன் .	உரிய காசோலையை ලැබුණ என்பதை அமைச்சு வங்கிக்கு அறியத் தருமாறு தயவுடன் அறியத் தருகிறேன் .
එල් . ජී . ධම්මිකා , හැරන්ඩල් බී , රඹුක්පිටිය .	எல் . ஜீ . ධම්මිකා , ஹெரெண்டெல் பீ .	எல் . ஜீ . தம்மிகா , ஹெரெண்டெல் பீ , .

From the experiment of PMI based chunking and baseline model, we could found out some observations. Those observations and analyze study of those observations are listed below.

- Translating Sandhi words correctly
    - நேரத் கொடுப்பனவு

Sandhi can be different according to next word. In baseline system, if the word is translated according to word alignment, it won't consider about next word. So in

baseline system, most of the times it doesn't consider about next word until the phrases are translating at a time. But when we chunk the collocation words, both words will be considered as a single word. So it helps with the proper Sandhi translation.

- Translating Conjunction words
    - பொ . க . கா . மற்றும் வியாபார அலுவலகம்

Conjunction words are used to join two words. In baseline system, if the word is translated according to word alignment, it won't consider about next words. So in baseline system, if the probability of word translation is higher than phrase translation, it will select word translation. So there is a chance of a missing translation of conjunction words. But when we chunk the collocation words, both words will be considered as a single word. So it helps for the conjunction translation.

- Translating not properly aligned words
    - முன்னுரிமை is translating in chunked based system. But it is not translating in baseline system.

In some cases, word alignment is not properly aligned within words. There can be some not properly aligned phrases as well. In that case, some words cannot be translated by the baseline system. But when we are chunking, those words are considered as the same word. So it is able to translate those words.

- Context awareness
    - மாறுபாடுகள் vs மாறிகள் in the sentence "ඉහත මාර්ගෝපදේශයට අනුව මෙම විචලන ලේඛන අනුමැතියට ප්‍රසම්පාදන කමිටුවට ඉදිරිපත් කිරීමට සිදු වේ ."

In this example, විචලන can be translated into மாறுபாடுகள் or மாறிகள். As we are chunking the words, it is translating the words according to the context rather than translating word by word.

- Mapping two words into one word
    - පාඩම් කරනවා (studies) translates into படிக்கிறார்

In this case, two words 'පාඩම්' and 'කරනවා' are merged into one word පාඩම්_කරනවා. So the translation system is able to translate those two words easily as படிக்கிறார்.

6.5.2 Results for NER based chunking and POS based chunking

In this subsection results of pre-processing models based on NER chunking and POS chunking are described. All the pre-processed models are evaluated with the same test-set which contains 300 Sinhala sentences. The well-known Machine Translation metrics BLEU is used to evaluate the system.

Her results of pre-processing based on NER chunking and pre-processing based on POS chunking are described. The results are compared with baseline system. The results are in terms of BLEU score and it is shown in Table 6.12.

Table 29 The results are in terms of BLEU score for all models

Models	**BLEU score**
Baseline	29.8
Pre-processing based on NER chunking	30.34
Pre-processing based on POS chunking	33.02
Hybrid model	33.41

Hybrid approach is done by combining preprocessing approaches based on PMI chunking, NER chunking and POS chunking. BLEU score was increased up to 33.41by using hybrid approach.

From the results in, it is shown that the proposed pre-processing method based on NER chunking improves the BLEU score compare to the baseline system. Likewise, pre-processing based on POS chunking also increases the BLEU score drastically from the baseline model. So both approaches perform better translation compare to the baseline system. Some sample translations generated by both systems are given in Table 6.13.

Table 30 Sample translations of both models

Input	Output(NER based)	Output(POS chunked)	Baseline system
පෙරුමාල් කිට්නසාමි , බාර්කේපල් මැද කොටස , නාවලපිටිය .	பெருமாள் கிட்னசாமி , பார்கேபல் මැද பிரிவு , நாவலபிட்டிய .	பெருமாள் கிட்னசாமி , ,பார்கேபல் மத்திய பகுதியின் , நாவலபிட்டிய .	பெருமாள் கிட்னசாமி , பார்கேபல் මැද பிரிவு , நாவலபிட்டிய .
ප්‍රමුඛතා කාර්යයක්	முன்னுரிமைப்படி	அடிப்படையிலும்	இருந்தும் செயல்

සේ සලකා මේ සම්බන්ධයෙන් ඔබ ලබා දෙන සහයෝගය බෙහෙවින් අගය කරමි .	மேற்கண்டவாறு கருதி இது தொடர்பாக நீங்கள் வழங்கும் ஆதரவை மிகவும் பாராட்டுகின்றேன் .	பணியினை கருதப்பட்டு இது தொடர்பாக நீங்கள் வழங்கும் ஆதரவை மிகவும் பாராட்டுகின்றேன் .	ஒன்றாகக் கருதி இது தொடர்பாக நீங்கள் வழங்கும் ஆதரவை மிகவும் பாராட்டுகின்றேன் .
එම තාක්ෂණ ඇගයීම් කමිටුවේ සංයුතිය පහත සඳහන් පරිදි වේ .	மேற்படி தொழில்நுட்ப மதிப்பீட்டுக் குழுவின் தொகுப்பு பின்வருமாறு .	அந்த தொழில் நுட்ப மதிப்பீட்டுக் குழுவின் தொகுப்பு பின்வருமாறு அமையும் .	அந்த தொழில் நுட்ப மதிப்பீட்டுக் குழுவின் தொகுப்பு பின்வருமாறு .

A hybrid approach was done by collaborating all these pre-processing based on chunking methods. In that approach, we have used top 500 and frequency=5 PMI based chunking, NER chunking and POS chunking. We could achieve BLEU score of 30.07. The value is lesser than the individual pre-processing method because when we integrate all the pre-processing techniques, the data got over chunked and got lesser BLEU score to compare to others. But even it is little higher than baseline system.

6.6 Preprocessing based on segmentation

The evaluation scores of the Sinhala to Tamil translation and the sample translations from the developed SMT with pre-processing based on segmentation are described in this section. The evaluation scores for the preprocessing techniques based on segmenting the words into subwords. We have trained the SMT without the pre-processing technique to compare the translation quality. The evaluation of translation quality is done by measuring the BLEU score of the translation of test data set. Even though these language resources are sparse, we have achieved much better BLEU score for the Sinhala to Tamil translation. Results obtained for the experiments are shown in Table 6.14.

Table 31 BLEU Score values of the traditional phrase-based and fully segmented approaches

Models	**BLEU score**
Baseline	29.8
Full segmentation, 3 gram	27.4
Full segmentation, 7 gram	30.16

By comparing the columns in Table 7.14, BLEU score value slightly increases when we increase language model size as 7-gram. Finally, the best BLEU score resulted from the fully-segmented approach with language model size 7-gram. However, it is not a significant improvement compared to the results of the baseline system.

When we consider the translated output, we can rarely see any not translated words, unlike in the baseline system. Even though the not translated words are rare, we could achieve only lower BLEU score values for the fully-segmented approach.

6.7 Tamil to Sinhala traditional SMT system

The evaluation scores of the Tamil to Sinhala translations and the sample translations from the developed traditional SMT are described in this section. Translation quality is measured by the BLEU score of test data set. Even though these language resources are sparse, we have achieved much better BLEU score for Tamil to Sinhala translation.

Our evaluated system's entire language model is from the inner domain and outer domain. And Number of 3-gram hit is relatively high in the language models. The best BLEU score we have got without tuning is 34.27. After the evaluation, we obtained highly excellent BLEU in Tamil to Sinhala System. The BLEU score results in high values which is a proof that the system has high accuracy. Compare to Sinhala to Tamil translation, Tamil to Sinhala translation gives better translation even if we use same parallel corpus because of Tamil language is morphologically rich compare to Sinhala language and 1 or 2 words in Tamil language is translated into one word in Sinhala language. It is easily said as many to one mapping is easier than one to many mapping.

The good thing about the research was that we had a very good score with normal settings, but more quality output can be expected after fine tuning the system. We used the Minimum Error Rate Tuning (MERT) technique to achieve more scores. After MERT technique we received new BLEU score as 35.01. This score shows some great improvements.

It is very much acceptable that MERT has enhanced the scores of the system at a good rate. When talking about the decision making of the system we have developed is that the scores are well-improved values than previous research on SMT with local languages like Tamil and Sinhala. And we can mark this as a successful research in the domain of official document.

6.7 Summary

This chapter described the results which are used to test the effectiveness of Sinhala to Tamil Machine translation systems. Initially, a study was carried out to find out the divergence between the Sinhala and Tamil languages and results are presented in this chapter. Then we have performed some pre-processing techniques to overcome those issues. The pre-processing techniques allow the developed system to achieve a relative improvement in BLEU score. The pre-processing techniques developed in this work helps to increase the translation quality. Within all pre-processing techniques, PMI based chunking gives better result compare to others. Even though Hierarchical phrase-based model didn’t give good result to Sinhala and Tamil translations, it gives better result to other language pairs.

CONCLUSION AND FUTURE WORK

This chapter summarizes the thesis, discusses its findings and contributions, a general conclusion based on the findings, points out limitations of the current work, and also outlines directions for future research. The conclusion, which follows after the summary, attempts to highlight the research contributions in the field of Sinhala and Tamil language translation processing. At the same time, the limitations and future scope of the developed systems are also mentioned, so that researchers who are interested in extending any of this work can easily explore the possibilities. The chapter is divided into three sections. Section 8.1 is a summary of the thesis. Section 8.2 brings the thesis to a conclusion and Section 8.3 discusses the limitations of the current work and future work.

7.1 Summary

Machine translation plays a vital role at present due to the multilingual nature of the current society. The necessity of machine translation arises with the need for translating resources of knowledge from one language to other increases. This research was focused on Sinhala-Tamil translation as Sinhala-Tamil translation gains importance since both Sinhala and Tamil are official languages practiced in Sri Lanka (along with English) although the majority of the population can read/write only one language. Further, since these two languages are considered as low resourced languages, these efforts gain more importance. The Sinhala language belongs to the Indo Aryan language family and the Tamil language belongs to the Dravidian family. As two languages that have been in contact for a long period of time, they share notable resemblances in morphology and syntax.

Currently, there are some automatic systems like “Si-Ta system” already available in the translation between the above languages. Most of those systems are based on traditional statistical machine translation system. So, there are some challenges in those systems to overcome for the better quality of the translation. The challenges are

- The divergence between the above languages when translating
- Quality of the translation
- Both languages are morphologically rich
- In the system, there is no translation from Tamil to Sinhala.

For a better translation, identifying the divergence between the languages is an important factor. So initially, we focused on language divergence between Tamil and Sinhala languages. Due to the results of the divergence, we have come up with the semi-automatic alignment algorithm for different POS tagsets and aligned the POS tagsets of Tamil and Sinhala languages. We have shown that the problem of heterogeneity in POS tagsets can be cast into the labeled tree alignment problem. We have presented a quality alignment between Sinhala UOM tagset and Tamil BIS tagset. We listed numerous examples from real tagsets of Tamil and Sinhala languages to illustrate the most difficult parts of tagsets alignment.

To overcome translation challenges such as reordering, Abbreviations and initials, word flow of the sentence, data sparseness and mapping one word with one or more words, we have the hierarchical phrase-based model and factored model. We have developed a hierarchical phrase-based SMT in the translations of Tamil to English, English to Tamil, Malayalam to English, English to Malayalam, Tamil to Sinhala and Sinhala to Tamil. The comparison between traditional SMT and HPM SMT for the translation between South Asian and English languages carried out in this study. The comparison between traditional SMT and HPM SMT for the translation between Sinhala and Tamil languages also carried out in this study.

We have developed factored model using Parts of speech knowledge in the translation of Sinhala to Tamil. Using the linguistic knowledge in SMT can reduce the need for massive amounts of data by raising the level of generalization, and thereby providing a basis for more efficient data exploitation. This is especially desirable for language pairs (like Sinhala and Tamil) where massive amounts of parallel corpora are not available. The comparison between traditional SMT and factored SMT for the translation between Sinhala and Tamil languages also carried out in this study.

This thesis presents the novel pre-processing methods to enhance the quality of the Sinhala to Tamil translation. PMI based collocation finding, POS-based chunking, NER based chunking and sub word construction (segmentation) are used to preprocess the corpus. These pre-processing techniques presented in this thesis can be applied directly to other language pairs especially for translating from morphologically rich language to another morphologically rich language. The precision of the translation system depends on the performance of techniques and tools used in the system. The

experimental results clearly demonstrate that the new techniques proposed in this research are definitely significant.

All these different machine translation models and preprocessing techniques have experimented and the BLEU scores are compared with the baseline system. Finally, this score is compared with "Google Translate" online machine translation system. It showed significantly good result compare to Google translate. Improvement in BLEU evaluation scores shows that this proposed approach is appropriate for Sinhala to Tamil Machine Translation system.

7.2 Conclusion

The major achievement of this research has been the improvement of translation between Sinhala and Tamil languages by preprocessing techniques. We observed all the preprocessing techniques include POS integration and segmentation outperform the baseline system for the translation between Sinhala-Tamil. It helps in reordering, better word choice, context awareness, translating conjunction/sandhi words, mapping one word with one or more words and transliterate some words. Within the all preprocessed methods PMI based chunking gave good result compare to other preprocessing techniques. Identifying language divergence and developing alignment of POS tagsets of Tamil and Sinhala languages are challenging and demanding tasks especially for morphologically rich languages like Tamil and Sinhala.

In this work we have developed a hierarchical phrase based SMT to resolve the issues translation challenges such as reordering, Abbreviations and initials, word flow of the sentence, data sparseness and mapping one word with one or more word. We have done hierarchical phrase based SMT of Tamil to English, English to Tamil, Malayalam to English, English to Malayalam, Tamil to Sinhala and Sinhala to Tamil. The comparison between baseline system and HPM SMT for the translation between South Asian and English languages carried out in this study. We observed HPM SMT outperform the baseline system for the translation between morphologically rich and poor languages (for the same dataset). Hierarchical phrase based models helps to improve translation quality between languages that vary by sentence structure. The comparison between baseline system and HPM SMT for the translation between Sinhala and Tamil languages also carried out in this study. However, in this case, traditional approach performs better compared to the hierarchical phrase model.

Hence, hierarchical phrase based models lower the quality of languages that share similar sentence structure since built in Parser is only available for English language in Moses tool.

As we didn't get results using hierarchical phrase based model, we have chosen to integrate linguistic features to the system. Because the performance of the statistical and machine learning methods mainly depends on the size and correctness of the corpus, if the corpus consists of all types of surface word forms, word categories and sentence structures, then it is possible for a learning algorithm to extract all required features. But both Sinhala and Tamil are low resourced languages. So, we have decided to add POS information to the corpora. We have developed a factored model using POS information. The comparison between baseline system and factored SMT for the translation from Sinhala to Tamil languages carried out in this study. We observed factored SMT outperform the baseline system for the translation between morphologically rich languages (for the same dataset). It helps in the reordering the sentences, better word choice, context awareness, translating conjunction words and transliterate some words.

Then we have used preprocessing techniques such as PMI based collocation finding, POS based chunking, NER based chunking and sub word construction (segmentation) towards addressing challenges such as unknown words, context awareness, better word choice, word flow, ambiguity in translation, translating proper Sandhi, translating name entities and mapping one word with one or more words. The comparison between baseline system and preprocessed SMT for the translation from Sinhala to Tamil languages carried out in this study. We observed all preprocessing techniques outperform the baseline system. Within the all preprocessed methods PMI based chunking gave good result compare to other preprocessing techniques. When we use hybrid model by using all these preprocessing based on chunking methods, BLEU score was increased up to 33.

8.2 Future Directions

The thesis addresses the technique to improve the quality of Machine Translation by factored model and preprocessing techniques. The main limitation of the approach presented here is that it is not directly applicable in the reverse direction (Tamil to Sinhala). All this developed preprocessing techniques and MT systems are domain

specific and scalable, so that researchers who are interested in extending any of this work can easily explore the possibilities. There are a number of possible directions for future work, based on the findings in this thesis. Some of the directions are given bellow.

- Increasing the size of parallel corpora always help to improve the accuracy of the system.
- The preprocessing techniques and methodologies which are developed are used to perform on translation between Sinhala to Tamil. It would be interesting to apply the similar methods for translating English to other morphologically rich languages.
- The preprocessing techniques and methodologies which are developed are can be used to develop a translation system that translate other languages into Tamil.
- Analyzing problems with existing data sets, the concern of morphology and its relation to output quality by combining those models together.
- Extending POS alignment for different tagsets which whether belong to different language or same language.
- Adding morphology as another factor in factored model

REFERENCES

[1] Hoang H, Birch A, Callison-Burch C, Federico M, Koehn P, "MOSES: Open Source Toolkit for Statistical Machine Translation.," In Proceedings of the 45th Annual Meeting of theACL on Interactive Poster and Demonstration Sessions, 2007, pp. 177-180.

[2] Bonnie J Dorr, "Machine translation divergences: A formal description and proposed solution," In Computational Linguistics, 1994, pp. 597-633.

[3] J Cocke, S Della Pietra, Brown P, "A statistical approach to machine translation," InJournal ofComputational Linguistics, vol. 16(2), pp. 79-85, 1990.

[4] Dzmitry, Kyunghyun Cho, Yoshua Bengio, Bahdanau, "Neural machine translation by jointly learning to align and translate," In arXiv preprint arXiv:1409.0473, 2014.

[5] Weerasinghe R, "A statistical machine translation approach to sinhala-tamil language translation," In Towards an ICT enabled Society, Sri Lanka, 2003, p. 136.

[6] Sripirakas A, Weerasinghe, Herath S, "Statistical machine translation of systems for Sinhala-Tamil," In International Conference in Advances in ICT for Emerging Regions (ICTer), 2010.

[7] R Weerasinghe, M Niranjan, R Pushpananda, "Sinhala-Tamil Machine Translation: Towards better Translation Quality," In Australasian Language Technology Association Workshop 2014, Melbourne, 2014.

[8] S Sheeyam, A Umasuthan, S Rajpirathap, "Real-time direct translation system for Sinhala and Tamil languages," In Computer Science and Information Systems (FedCSIS), 2015.

[9] R Weerasinghe M Niranjan, R Pushpananda, "Statistical Machine Translation from and into Morphologically Rich and Low Resourced Languages," In International Conference on Intelligent Text Processing and Computational Linguistics, 2015.

[10] Rosenfield, Clarkson P R, "Statistical Language Modelling using the CMU-Cambridge Toolkit," In Proceedings ESCA Euro Speech, Greeze, 1997.

[11] S Roukos, T Ward, W J Zhu, K Papineni, "BLEU: a method for automatic evaluation of machine translation.," In Proceedings of the 40th annual meeting on association for computational linguistics, 2002.

[12] WSWS. (2018, August) World Socialist Web Site. [Online]. www.wsws.org

[13] University of Colombo. (2018, August) Language Technology Research Laboratory. [Online]. http://www.ucsc.cmb.ac.lk/ltrl/

[14] Stolcke A et al., "SRILM-An Extensible Language Modeling Toolkit," In INTERSPEECH, 2002.

[15] Och FJ, "Minimum Error Rate Training in Statistical Machine Translation," In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, 2003, pp. 160-167.

[16] Pears R, Koh Y S, Sakthithasan S, "One pass concept change detection for data streams.," In Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, 2013, pp. 461-472.

[17] Mahendran Jeyakaran, Ruvan Weerasinghe, "A novel kernel regression based machine translation system for sinhala-tamil translation," In Proceedings of the 4th Annual UCSC Research Symposium, 2011.

[18] Lagus K, Creutz M, "Unsupervised Models for Morpheme Segmentation and Morphology Learning," In ACM Transactions on Speech and Language Processing (TSLP), 2007, p. 4.

[19] Hoang H, Birch A, Callison-Burch C, Federico M, Koehn P et al., "MOSES: Open Source Toolkit for Statistical Machine Translation.," In Proceedings of the 45th Annual Meeting of theACL on Interactive Poster and Demonstration Sessions, 2007, pp. 177-180.

[20] Ranathunga S, Jayasena S, Dias G, Farhath F, "Integration of Bilingual Lists for Domain-Specific Statistical Machine Translation for Sinhala-Tamil," In Moratuwa Engineering Research Conference (MERCon, Moratuwa, 2018, pp. 538-543.

[21] Pranavan Theivendiram, Surangika Ranathunga, Sanath Jayasena, Gihan Dias, Fathima Farhath, "Improving Domain-specific SMT for Low-resourced Languages using Data from Different Domains," In 11th Edition of the LAnguage Resources and Evaluation Conference, 2018.

[22] Prabath Sandaruwan, Malith Thilakarathne, Achini Herath, Surangika Ranathunga, Sanath Jayasena, Gihan Dias, Pasindu Tennage, "Neural Machine Translation for Sinhala and Tamil Languages," In International Conference on Asian Language Processing (IALP), 2017, pp. 189-192.

[23] Y Kim, Y Deng, J Senellart, A. M. Rush, G Klein, "OpenNMT: Open-source toolkit for neural machine translation," In arxiv preprint arXiv:1701.02810 [cs.CL], 2017.

[24] Bonnie J, Dorr, "Machine translation divergences: A formal description and proposed solution," In Computational Linguistics, 1994, pp. 597-633.

[25] Khan M A, Saboor A, "Lexical-semantic divergence In Urdu-to-English example based Machine Translation.," In 6th International Conference In Emerging Technologies (ICET), 2010, October, pp. 316-320.

[26] Deshmukh P D, Kazi, M M Kale, Kulkarni S B, "Linguistic Divergence Patterns in English to Marathi Translation," In International Journal of Computer Applications, 2014.

[27] Niladri Sekhar, Dash, "Linguistic Divergences in English to Bengali Translation," In International Journal of English Linguistics, 2013.

[28] T M Kirby, Ellison, "Measuring language divergence by intra-lexical comparison," In Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics, 2006, pp. 273-280.

[29] Mourya N, Pandey V, Behera P, "Dealing with Linguistic Divergences in English-Bhojpuri Machine Translation.," In Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016), 2016, pp. 103-113.

[30] Tampoe H.D, "Sinhala and Tamil: A case of Contact-Induced Restructuring," Language and Linguistics, Newcastle University, A PhD Dissertation presented at School of English Literature http://hdl.handle.net/10443/3552, 2016.

[31] R McDonald, "Universal Dependency Annotation for Multilingual Parsing.," In Proceedings of ACL, Sofia, Bulgaria, 2013.

[32] M Anand Kumar, "Factored Statistical Machine Translation System for English to Tamil Language.," Pertanika Journal of Social Sciences & Humanities 22.4, 2014.

[33] M Anand Kumar, "Morphology based Prototype Statistical Machine Translation System for English to Tamil Language," AMRITA School of Engineering, AMRITA Vishwa Vidyapeetham, Coimbatore, A Thesis Submitted for the Degree of Doctor of Philosophy in the School of Engineering 2013.

[34] Bureau Indian Standard, "Unified Parts of Speech (POS) Standard in Indian Languages".

[35] Sudhakar Kumawat, Vinayak Srivastava, Nitish Chandra, "Various Tagsets for Indian Languages and Their Performance In Part Of Speech Tagging.," In Proceedings of 5th IRF International Conference, ISBN: 978-93-82702-67-2, 2014.

[36] Ramanathan M V, "An Attempt at Multilingual POS Tagging for Tamil.".

[37] IIIT. (2018) IIIT HYDERABAD. [Online]. http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines

[38] Lakshmana Pandian S T, "Morpheme based Language Model for Tamil Part-of-Speech Tagging.," In Polibits 38, 2008, pp. 19-25.

[39] Natarajan A M, Selvam M, "Improvement of rule based morphological analysis and pos tagging in tamil language via projection and induction techniques.," In International journal of computers, 2009, pp. 357-367.

[40] Central Institute of Indian Languages. (n.d.). (2018, August) Central Institute of Indian Languages. [Online]. http://www.ciil.org/

[41] Anand Kumar, Shivapratap G, Soman K P, Rajendran S, Dhanalakshmi V, "Tamil POS Tagging using Linear Programming," In International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May, 2009.

[42] Anand kumar M, Rajendran S, Soman K P, Dhanalakshmi V, "POS Tagger and Chunker for the Tamil Language.," In Proceedings of Tamil Internet Conference., 2009.

[43] Vasu Renganathan, "Development of Part-of-Speech Tagger for Tamil," In Tamil Internet 2001 Conference August, Kuala Lumpur, 2001, pp. 26-28.

[44] Ganesan M, "Morph and POS Tagger for Tamil (Software)," Annamalai University , Annamalai Nagar, 2007.

[45] Rajendran S, "Parsing in Tamil," in LANGUAGE IN INDIA www.languageinindia.com Volume 6: 8 August, 2006.

[46] AUKBC. (2018, August) AU-KBC RESEARCH CENTRE. [Online]. http://www.au-kbc.org/research_areas/nlp/projects/postagger.html

[47] Sobha L, Kumara Shanmugam B, Arulmozhi P, "Parts of Speech Tagger for Tamil," In Proceedings of the Symposium on Indian Morphology,Phonology & Language Engineering, Indian Institute of Technology, Kharagpur., 2004.

[48] Sobha, L Arulmozhi, "A Hybrid POS Tagger for a Relatively Free Word Order Language," In Proceedings ofMSPIL-2006, Indian Institute of Technology, Bombay., 2006.

[49] AUKBC. (2018, August) AUKBC Research Centre. [Online]. http://www.au-kbc.org/

[50] W V Welgama, A R Weerasinghe, Dilmi Gunasekara, "Hybrid Part of Speech Tagger for Sinhala Language.," In International Conference on Advances in ICT for Emerging Regions (ICTer)., 2016, pp. 041 – 048.

[51] Ranathunga S, Jayasena S, Dias G, Fernando S, "Comprehensive Part-Of-Speech Tag Set and SVM Based POS Tagger for Sinhala.," In WSSANLP 2016, 2016, p. 163.

[52] N G J Dias, Jayaweera A J P M P, "Hidden Markov Model Based Part of Speech Tagger for Sinhala Language," In arXiv preprint arXiv:1407.2989, 2014.

[53] A R Weerasinghe, Jayasuriya M, "Learning a stochastic part of speech tagger for sinhala," In 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), 2013.

[54] Wilson A, Leech G, "Recommendations for the Morphosyntactic Annotation of Corpora.," EAGLES Re-port EAG-TCWG-MAC/R, 1996.

[55] Kalika Bali, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Girish Nath Jha, Rajendran S, Saravanan K, Sobha L, Sankaran Baskaran, "Designing a Common POS-Tagset Framework for Indian Languages," In The 6th Workshop on Asian Languae Resources, 2008.

[56] Norbert, Suzanne Lenz, Volz, "Multilingual Corpus Tagset Specifications," In MLAP PAROLE 63œ386 WP 4.4, 1996.

[57] Nancy Ide, Jean, Véronis, "Multext (multilingual tools and corpora)," In Proceedings of the 15th International Conference on Computational Linguistics (COLING -94), Kyoto, Japan, 1994.

[58] Tomaž, Erjavec, "Multext-east version 3: Multilingual morphosyntactic specifications, lexicons and corpora.," In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004), Lisboa, Portugal, 2004.

[59] D Das, Petrov, "Unsupervised part-of-speech tagging with bilingual graph-based projections," In Proceedings of ACL-HLT, 2011.

[60] B Snyder, J Eisenstein, R Barzilay, T Naseem, "Multilingual part-of-speech tagging: Two unsupervised approaches," In JAIR, 36, 2009.

[61] Zeman D, "Reusable Tagset Conversion Using Tagset Drivers.," In Proceedings of LREC, Marrakech, Morocco, 2008.

[62] Tyers, Joakim Nivre, Francis M, "Universal Dependencies for Turkish.," In Proceedings of COLING, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 2016, pp. 3444–3454.

[63] Timothy Dozat, Natalia Silveira, Katri Haverinen, FilipGinter, Joakim Nivre, Christopher D, Marie Catherine De Marneffe, "Universal Stanford Dependencies: a crosslinguistic typology," In Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC), Reykjav´ ık, Iceland, 2014.

[64] Reut, Tsarfaty, "A unified morpho-syntactic scheme of Stanford Dependencies.," In Proceedings of 51st Annual Meeting of the Association for Computational Linguistics (ACL), 2013, pp. 578-584.

[65] Alon Y, Halevy, AnHai Doan, "Semantic-Integration Research In the Database Community A Brief Survey.," AI Magazine Volume 26 Number 1, 2005.

[66] C Lenzerini, M Navathe, Batini, "A Comparative Analysis of Methodologies for Database Schema Integration.," In ACM Computing Survey 18(4), 1986, pp. 323–364.

[67] Chiang D, "A hierarchical phrase-based model for statistical machine translation," In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 2005, June, pp. 263-270.

[68] Aho J D, Ullman A V, "Syntax directed translations and the pushdown assembler," Journal of Computer and System Sciences, pp. 3:37–56., 1969.

[69] Franz Josef Och, Hermann, Ney, "Improved statistical alignment models," In Proceedings of the 38th Annual Meeting of the ACL, 2000, pp. 440-447.

[70] Franz Josef Och, Daniel Marcu, Philipp Koehn, "Statistical phrase-based translation," In Proceedings of HLT-NAACL, 2003., pp. 127-133.

[71] Philipp Koehn, "Pharaoh: a beam search decoderfor phrase-based statistical machine translation models," In Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas, 2004, pp. 115-124.

[72] Mahsa, Abdolhossein Sarrafzadeh, Mohaghegh, "A hierarchical phrase-based model for English-Persian statistical machine translation," In 2012 International Conference on Innovations in Information Technology (IIT), 2012.

[73] Zeman D, Jawaid B, "Word-order issues in English-to-Urdu statistical machine translation," In The Prague Bulletin of Mathematical Linguistics, 2011, pp. 87-106.

[74] Nadeem, Khan et al., "English to Urdu hierarchical phrase-based statistical machine translation," In Proceedings of the 4th Workshop on South and Southeast Asian Natural Language Processing, 2013.

[75] Ulrich, Germann, "Building a statistical machine translation system from scratch: how much bang for the buck can we expect?," In Proceedings of the workshop on Data-driven methods in machine translation-, 2001, p. Volume 14.

[76] Vasu, Renganathan, "An ineractive approach to development of English to Tamil machine translation system on the web," In Proceedings of INFITT, 2002.

[77] Loganathan R M, "English-Tamil Machine Translation System," Amrita Vishwa Vidyapeetham, Coimbatore, Master of Science by Research Thesis 2010.

[78] Dhanalakshmi V, Soman K P, Sharmiladevi V Kumar M, "Improving the Performance of English-Tamil Statistical Machine Translation System using Source-Side Pre-Processing.," In arXiv preprint arXiv:1409.8581, 2014.

[79] Mary Priya, G Santhosh Kumar, Sebastian, "English to malayalam translation: a statistical approach," In Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India. ACM, 2010.

[80] Daniel Marcu, William, Wong, "A Phrase-Based, Joint Probability Model forStatistical Machine Translation," In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), Philadelphia, 2002.

[81] Kay, Stephan Vogel, Rottmann, "Word reordering in statistical machine translation with a POS-based distortion model," In Proc. of TMI, 2007, pp. 171-180.

[82] Deepa, Mauro Cettolo, Marcello Federico, Gupta, "POS-based reordering models for statistical machine translation," In Proceedings of the Machine Translation Summit (MT-Summit), 2007.

[83] Derek F, Wong, Lidia S, Chao, Li Shuo, "Experiments with POS-based restructuring and alignment-based reordering for statistical machine translation," In Proceedings of the Second Workshop on Hybrid Approaches to Translation, 2013.

[84] Wetzel D, Kaeshammer M, "Enriching phrase-based statistical machine translation with POS information." In Proceedings of the Second Student Research Workshop associated with RANLP, 2011, pp. 33-40.

[85] N Dhanesh, "Conceptual Framework for Automated English to Tamil Machine Translation System," Scholar, P. G. A.

[86] Ney H, Ueffing N, "Using pos information for statistical machine translation into morphologically rich languages," In Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1, 2003, April, pp. 347-354.

[87] Philipp Koehn, Hieu, Hoang, "Factored translation models," In Proc EMNLP+CoNLL, Prague, 2007, pp. 868–876.

[88] Sonja Nießen, Hermann, Ney, "Statistical Machine Translation with Scarce Resources Using Morpho-syntactic Information," In Journal ofComputational Linguistics, 30(2), 2004, pp. 181–204.

[89] A Potamianos, I Klasinas, P Karageorgakis, "Towards incorporating language morphology into statistical machine translation systems," In Proceedings Automatic Speech Recogn. and Underst. Workshop (ASRU), 2005.

[90] Soha, Sultan, "Applying morphology to English-Arabic statistical machine translation.," Diss. Master's Thesis Nr. 11 ETH Zurich In collaboration with Google Inc. 2011.

[91] Adria de Gispert, Ramis, "Introducing Linguistic knowledge into statistical Machine Translation," TALP Research Center, Speech Processing Group

Department of Signal Theory and Communications, Universitat Polit`ecnica de Catalunya., Ph.D. thesis 2006.

[92] Sara, Stymne, "Compound Processing for Phrase-Based Statistical Machine Translation," Linköping University, Sweden, Licentiate thesis 2009.

[93] Rabih M, Zbib, "Using Linguistic Knowledge in Statistical Machine Translation," Massachusetts Institute Of Technology, Ph.D. thesis September, 2010.

[94] Lee Y S, "Morphological analysisfor statistical machine translation," Defense Technical Information Center, 2004.

[95] S S Arora, K Agrawal, "Pre-processing English-Hindi Corpus foStatistical Machine Translation," In Computación y Sistemas, 2017, p. 21(4).

[96] Chengqing Zong ,Bo Xu, Yu Zhou, "Bilingual Chunk Alignment In Statistical Machine Translation," In IEEE International Conference on Systems, Man and Cybernetics, 2004.

[97] Santanu, Sudip Kumar Naskar, Sivaji Bandyopadhyay, Pal, "Word Alignment-Based Reordering of Source Chunks in PB-SM," In LREC, 2014.

[98] Arianna, Daniele Pighin, Marcello Federico, Bisazza, "Chunk-lattices for verb reordering in Arabic–English statistical machine translation," In Machine translation 26.1-2 (2012), 2012, pp. 85-103.

[99] Xiaolin, Wang X, Utiyama M, Finch A, Sumita E, Wang, "Refining word segmentation using a manually aligned corpus for statistical machine translation," In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1654-1664.

[100] Anoop Kunchukuttan, Pushpak Bhattacharyya, Raj Dabre, More Rohit, "Augmenting Pivot based SMT with word segmentation," In Proceedings of the 12th International Conference on Natural Language Processing, 2015, pp. 303-307.

[101] Peyman, Qun Liu, Andy Way, Passban, "Providing Morphological Information for SMT Using Neural Networks," In The Prague Bulletin of Mathematical Linguistics 108, 2017, pp. 271-282.

[102] Wikipedia. (2018, August) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Tamil_language

[103] Wikipedia. (2018, August) Wikipedia. [Online]. https://en.wikipedia.org/wiki/Sinhalese_language.

[104] Robert, Caldwell, "A comparative grammar of the Dravidian or South-Indian Family of Languages," London, 1875.

[105] S Rajendran. (2006, Volume 6 : 8.) Language in India. [Online]. www.languageinindia.com

[106] Ramakrishnan S, "kiriyAvin thaRkAla thamiz akarAthi," Cre-A, 2006.

[107] Vilar, D Ney, H Jovicic S, Saric Z, Popovic M, "Augmenting a Small Parallel Text with Morpho-Syntactic Language Resources for Serbian-English Statistical Machine Translation.," In Proceedings of the ACL Workshop on Building and Using Parallel Texts, 2005, pp. 41–48.

[108] El-Kahlout I D, Oflazer K, "Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation," In: Proceedings of the Second Workshop on Statistical Machine Translation, 2007, pp. 25–32.

[109] Segalovich I, "A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine." In In: MLMTA, CiteSeer, 2003, pp. 273–280.

[110] V ayrynen, J J Creutz, Sadeniemi M, Virpioja S, "Morphology-Aware Statistical Machine Translation based on Morphs Induced in an Unsupervised Manner," In Machine Translation Summit XI, 2007, pp. 491–498.

[111] Jakob, Elming, Copenhagen Business School, PhD Thesis. 2008.

[112] George, Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics.," In Proceeding of the ARPA Workshop on Human Language Technology., 2002.

[113] Levenshtein V I, "Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady," , 1966 February, pp. 707–710.

[114] Stefan Vogel, Hermann Ney, Alex Zubiaga, Christoph Tillmann, "A DP-based search using monotone alignments in statistical translation.," In Proceedings of the 35th Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association forComputational Linguistics, Somerset, New Jersey, 1997, pp. 289–296.

[115] B Dorr, R Schwartz, L Micciulla, J Makhoul, Snover M, "A Study of Translation Edit Rate with Targeted Human Annotation" In Proceedings of Association for Machine Translation in the Americas, 2006, pp. 223-231.

[116] F J Och, "An Efficient method for determining bilingual word classes," In Proceedings of Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL), 1999.

[117] Heafield K, "KenLM: Faster and smaller language model queries.," In Proceedings of the Sixth Workshop on Statistical Machine Translation, 2011.

[118] Huang, Chiang L, "Forest rescoring: Faster decoding with integrated languagemodels," In Annual Meeting-Association For Computational Linguistics., 2007.

[119] NLTK. (2018, May) NLTK 3.3 documentation. [Online]. http://www.nltk.org/howto/collocations.html

[120] Pavel, Pecina, "An Extensive Empirical Study of Collocation Extraction Methods," In Proceedings of the Association for Computational Linguistics Student Research Workshop, 2005, pp. 13–18.

[121] P Hanks, W K Church, "Word association norms, mutual information and lexicography," In Proceedings of the 27th meeting of the Association of Computational Linguistics, 1989, pp. 76-83.

[122] Pathirennehelage N, Ihalapathirana, A Mohamed, M Z Ranathunga, S Jayasena, Dias G, Fernando S, Hameed R A, "Automatic Creation of a Sentence Aligned Sinhala-Tamil Parallel Corpus.," In Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing(WASSANLP), 2016, pp. 124-132.