Sinhala - Tamil Statistical Machine Translation (SMT) for Official Documents

Farook Fathima Farhath

(168034C)

Degree of Master of Philosophy in Computer Science and Engineering

Department of Computer Science And Engineering

University of Moratuwa
Sri Lanka

Oct 2018

Sinhala - Tamil Statistical Machine Translation (SMT) for Official Documents

Farook Fathima Farhath

(168034C)

Thesis submitted in partial fulfillment of the requirements for the
Degree of Master of Philosophy in Computer Science and Engineering

Department of Computer Science And Engineering

University of Moratuwa
Sri Lanka

Oct 2018

Declaration

I, Farook Fathima Farhath, declare that this is my own work and this thesis does not incorporate without acknowledgement 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).

Signed: ______________________________

Date: ______________________________

The above candidate has carried out research for the MPhil thesis under my supervision.

Name of Supervisor: Prof. Sanath Jayasena

Signature of supervisor: ______________________ Date: __________

Name of Supervisor: Dr. Surangika Ranathunga

Signature of supervisor: ______________________ Date: __________

Abstract

Sinhala and Tamil are declared to be the official languages of Sri Lanka. This requires each government related dissemination/communication to be done in both the languages. Even though the requirement for translation is higher, the number of available human translators is limited. One feasible option to boost the productivity would be assisting the human translators with machine translation output. Here the machine translation output is given to translators to work on by post editing, rather than translating from the scratch. However, Sinhala - Tamil pair does not have any well-performing machine translation system. Therefore, the focus of this research is to develop a machine translation system for short official government documents.

This thesis presents two main contributions towards building 'Si-Ta', the first domain-adapted machine translation system for Sinhala - Tamil. The first contribution is building the baseline translation system. The second is implementing data pre-processing techniques to improve the translation quality of the baseline system.

The baseline system was built using Moses, a phrase-based statistical translation system. This was the feasible option with the available resources.

To improve the quality of the translation, three main approaches were explored. They are: (a) domain adaptation, (b) integration of terminology, dictionary, and name lists, and (c) addressing out-of-vocabulary (OOV) problem using word-embedding-based paraphrasing.

In order to adapt the system for the domain of official government documents, different language model design techniques and a data filtration technique were experimented. Under terminology integration, experiments were carried out to evaluate the effect of incorporating bilingual terminology lists to the system. Moreover, a novel data augmentation technique was experimented to generate parallel data using bilingual lists and available parallel data. Further, open domain dictionary entries, as well as a list of person names and addresses were integrated and evaluated. In addition, word-embedding-based paraphrasing was used along with a novel heuristic-based filtering to address the out-of-vocabulary issue.

All the above-mentioned approaches gave an improvement over the baseline, apart from data filtering technique. Yet, all these scores were above the scores of already available machine translation systems for this language pair. Though our techniques/approaches were evaluated only on Sinhala - Tamil pair, they are feasible to be applied to other low-resourced, highly inflectional language pairs.

Keywords: Machine Translation, Sinhala, Tamil, Domain Adaptation, Terminology Integration, Out-of-vocabulary

Acknowledgements

First and foremost, I would like to extend my heartfelt gratitude to my supervisors, Dr. Surangika Ranathunga and Prof. Sanath Jayasena for their great guidance, mentoring, suggestions and their patience over past 2 ½ years. Their worthy feedback and comments helped me a lot in carrying out this research in a successful manner.

Beside my supervisors, I would like to extend my thanks to the members of the NLP Center of the University of Moratuwa. Especially, Dr. Uthayasanker Thayasivam and Prof. Gihan Dias for the assistance and guidance and worthy feedback that helped me with this research. And would much appreciate the precious times spent with my research colleagues.

I would like to acknowledge various bodies, funded my research. Especially, the Senate Research Council (SRC) Grant of the University of Moratuwa and the Department of Official Language, Sri Lanka.

Also, I would extend my gratitude to the Head of the Department and the staff of Department of Computer Science and Engineering for the various supports rendered by them and for giving me the opportunity to pursue this degree.

Further, I am in much debt to my friends who lend a helping hand in various means. Especially their motivating words were a comfort at the times when I needed sympathy.

Finally, I would like to extend thanks to the warriors behind the screen, my husband Azaam, son Rashaadh, my parents and siblings for their great support, commitment, encouragement, and dedications which made this dream a reality. They have dedicated their own time from their busy schedules to support me. And especially I regret and thank Azaam and Rashaadh for being patient at busy times.

Contents

List of Figures

List of Tables

List of Abbreviations

BLEU	**B**i**L**ingual **E**valuation **U**nderstudy
CSLM	**C**ontinuous **S**pace **L**anguage **M**odel
DMT	**D**irect **M**achine **T**ranslation
EBMT	**E**xample **B**ased **M**achine **T**ranslation
EM	**E**xpectation **M**aximization
HMM	**H**idden **M**arkov **M**odel
HTER	**H**uman-targeted **T**ranslation **E**dit **R**ate
LM	**L**anguage **M**odel
MT	**M**achine **T**ranslation
MTEOR	**M**etric for Evaluation of **T**ranslation with **E**xplicit **O**rde**R**ing
MWE	**M**ulti **W**ord **E**ntity
NE	**N**amed **E**ntity
NER	**N**amed **E**ntity **R**ecognizer
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NLP	**N**atural **L**anguage **P**rocessing
NMT	**N**eural **M**achine **T**ranslation
OOV	**O**ut **O**f **V**ocabulary
PBSMT	**P**hrase **B**ased **S**tatistical **M**achine **T**ranslation
POS	**P**arts **O**f **S**peech
RBMT	**R**ule **B**ased **M**achine **T**ranslation
TBMT	**T**ransfer **B**ased **M**achine **T**ranslation
SMT	**S**tatistical **M**achine **T**ranslation
TER	**T**ranslation **E**dit **R**ate

TM	**T**ranslation **M**odel
WER	**W**ord **E**rror **R**ate

Chapter 1

Introduction

Machine translation (MT) is the process of automatic translation of text from one natural language to another. Research on MT has been done from the early times when the computers were started to be used for commercial purposes. The initial workable solution was available in the early 1970s. Currently, many MT systems are being used as online translators and as an assistant in manual translation workflows for many language pairs [2]. However, only minimal research has been done for Sinhala and Tamil, the local languages of Sri Lanka. This thesis presents our work on building a domain-specific MT system for Sinhala and Tamil, a low-resourced language pair.

1.1 Problem Definition

Sri Lanka is a multi-ethnic country where the two languages, Sinhala and Tamil are declared as the official languages. Therefore the citizens of Sri Lanka must be able to communicate with the government in either of these official languages. However, only a very few people are fluent in both the languages. Therefore, government offices/agencies have to send letters to the citizens in the language the latter understands. This language barrier has been one cause behind the 30-year long civil war in the country that ended in 2009. In order to overcome this language barrier, currently, the assistance of human translators is used. In most scenarios, the original Sinhala documents are translated into Tamil before publishing. Yet the demand for human translators is much higher than the supply, which leads to many issues including the delay in publishing. This fails the goal of facilitating the citizens to use the language of their preference in official communication with the government. To move forward towards better bilingual communication between the government and the public, one solution is to boost the efficiency of human translators. Previous studies for European language pairs show that this can be achieved by introducing Machine Translation into the translation workflow [2].

1.2 Motivation

Currently, for many language pairs (European languages, Arabic, Chinese, etc.), domain-specific machine translations are being used successfully in various domains/fields such as medical, military, manufacturing and administration [2]. Yet, for Sinhala and Tamil, none of such systems are available apart from the openly available Google Translate [1], which is an open domain system. While the quality of its translation is poor [3], the confidentiality of the documents is also an issue when using open systems to translate official documents.

Therefore, to address these above-mentioned challenges, the ideal solution would be to develop a domain-specific MT system for this language pair. However, this language pair is low-resourced and highly inflected. They lack reasonably sufficient parallel data as well as well-performing linguistic tools [4]. These shortcomings result in a poor quality system that will not be deployable in a production workflow. Further, these will fuel up common challenges such as higher rate of out-of-vocabulary (OOV) words, poor output fluency, and inferior multi-word expression translation. Therefore, in order to develop a reasonably well performing system with the available resources, extra attention is required.

[1] https://translate.google.com/

1.3 Objective of Research

The aim of this research is to implement and improve a machine translation system for Sinhala and Tamil languages, for the domain of short official government documents, by incorporating effective pre-processing techniques. The research and development task can be split into sub-components as follows:

- Develop a baseline Sinhala - Tamil machine translation system.
- Adapting the system to the domain of short official government documents.
- Develop techniques to improve the baseline system by solving problems such as OOV.

1.4 Research Contributions

The contributions of this thesis are as follows:

- Development of ‘Si-Ta’ system (Sinhala - Tamil and Tamil - Sinhala machine translation system), which is currently backing the translation portal used by the translators in the Department of Official Languages.
- A novel data augmentation technique to generate parallel data out of a bilingual list, which can be used for parallel sentence generation for low-resourced setups.
- A novel paraphrase technique based on word-embedding and simple heuristics, which is feasible for a low-resourced and highly inflected language pair with agglomerate nature.

1.5 Publications

The thesis is based on the author’s contribution to the following three publications:

- Fathima Farhath, Theivendiram Pranavan, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. Improving Domain-specific SMT for Low-resourced Languages using Data from Different Domains. In *Eleventh International Conference on Language Resources and Evaluation (LREC) (2018).* Miyazaki, Japan.

- Fathima Farhath, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. Integration of Bilingual Lists for Domain-Specific Statistical Machine Translation for Sinhala-Tamil. In proceedings of *2018 Moratuwa Engineering Research Conference (MERCon)*, pages 538-543. IEEE, 2018.

- Surangika Ranathunga, Fathima Farhath, Uthayasanker Thayasivam, Sanath Jayasena and Gihan Dias. Si-Ta: Machine Translation of Sinhala and Tamil Official Documents. In *National IT Conference (NITC) (2018).* Colombo, Sri Lanka (in press).

1.6 Outline

The rest of the thesis is organized as follows:

Chapter 2 presents the literature on prior work on topics related to this research, especially a detailed description of SMT, as the system is based on SMT. It also includes a sufficient description of existing research for domain adaptation, terminology integration, dictionary integration and out-of-vocabulary handling techniques.

Chapter 3 details the work done on building the baseline ‘Si-Ta’ SMT system.

Chapter 4 explains the experimental procedures carried on top of the baseline system in order to improve the translation quality. This includes domain adaptation, terminology integration, dictionary integration, and out-of-vocabulary handling.

Chapter 5 presents the views on the results of the experiments that were carried on.

Chapter 6 presents the conclusion along with the future avenues on this research.

Chapter 2

Literature Review

This chapter gives a general introduction to Machine Translation (MT), history and different MT implementations. As the thesis is based on Statistical Machine Translation (SMT); more focus is given to the fundamental details of SMT, where details on the concepts, models and tools deployed are discussed. Next, the chapter covers details on the languages of interest (Sinhala and Tamil), the previous research carried on for this language pair and current status. The latter part of the chapter discusses the past research on various techniques on improving the quality of SMT output.

2.1 Machine Translation History

Machine translation is a sub-branch under Natural Language processing (NLP), where the computers are used to partially or fully automate the process of translation from a natural language to another [5].

The motivation of using machine translation is to speed up the process of translation without or with less involvement of human. MT plays a major role in breaking the language barrier between people. The application of MT can be broadly divided into three major categories as [6]:

1. **Assimilation**: Translation of a foreign material for the purpose of understanding (e.g. Attempt to understand articles such as news, blog articles, product information/reviews that are written in different languages). Most of the times the online translation engines such as Google Translator[1] and Bing Translator [2] are used.

2. **Dissemination**: Content in one language is translated into other languages in order to be published. E.g. Publishing notices/advertisements in multiple languages. Final output needs to be precise in quality. MT systems are integrated into professional translation workflow. The MT is used to provide intermediate versions for the human to post-edit.

3. **Communication**: Lowering the language barrier between people when a common language is not known. Instances are emails and chat rooms.

Research on machine translation was conducted from an earlier time when computers were started to be used for commercial purposes. However, the initial attempts were halted due to the negative assessments given by the Automatic Language Processing Advisory Committee (ALPAC) in 1966. This stopped almost all research funding from the US agencies towards MT. The report showed that the cost of MT is much higher than the human translation, and the requirement of translation automation was not a need of great necessity. The suggestion was to direct the funding to basic linguistic research [7].

Even though the US funding was reduced in MT research, Europe and other countries were active in the track; funded by government and commercial companies. However, the first translation company Systran was founded in 1968 by the Georgetown University (US) [2]. Thereafter many companies, as well as universities, initiated research on MT. Currently there are many successful deployments of MT systems for many language pairs [2].

[1] translate.google.com/

[2] https://www.bing.com/translator

2.2 Different Approaches

The approaches to MT can be mainly categorized into two, such as rule-based (linguistic) and corpus-based (data-driven) [6], based on the core methodology that is being used.

2.2.1 Rule-Based Machine Translation (RBMT)

Rule-Based Machine Translation (RBMT), also known as Knowledge-Based Machine Translation is the initial methodology used in implementing systems. Translation is done based on a set of rules set up with linguistic analysis such as morphological, syntactic and semantic analysis, on either language. Based on how deep the analysis is done towards an intermediate language-independent representation, the approaches are broadly divided into three categories as Direct, Transfer-Based and Interlingual. Vauquois Triangle [1] illustrates the levels of analysis as shown in Figure 2.1.

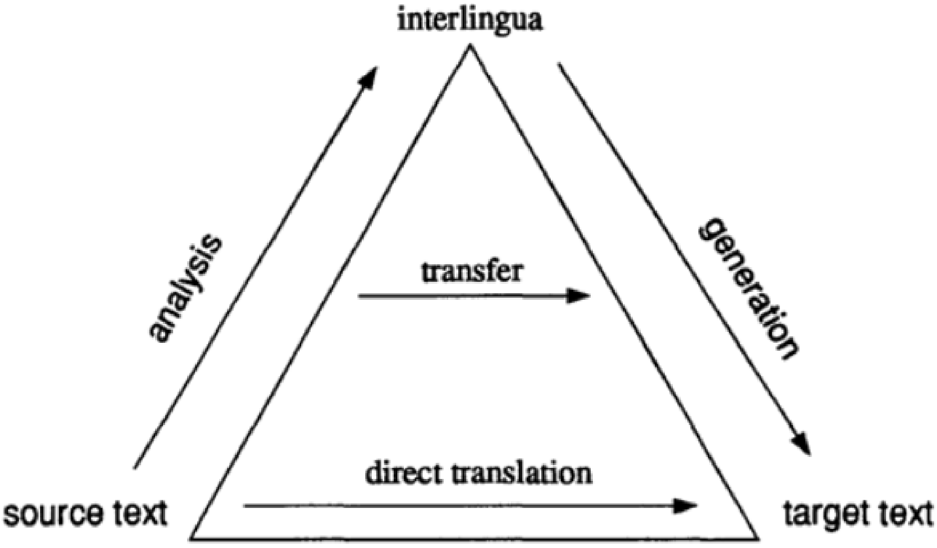


FIGURE 2.1: Vauquois Triangle (source: [1])

2.2.1.1 Direct Machine Translation (DMT)

In Direct Machine Translation, the source language is directly transformed to the target languages without having any intermediate representations. This is the approach used in the initial MT systems. This implementation uses simple grammar rules over bilingual dictionaries most of the time [8]. As the translation is done word by word, the depth of the analysis is less when compared to the other approaches. Therefore, the quality of the output is low at the lexical level. Also, the semantic quality of the translation is low where most of the time the translations are incorrect. The implementation does analysis of a source language towards a target language. Therefore, the systems are highly coupled between the language pair as well as to the direction.

2.2.1.2 Transfer-Based Machine Translation Approach (TBMT)

In Transfer-Based Machine Translation (TBMT), an intermediate representation of the source, as well as target languages is generated at the time of translation. The translation process includes steps of analysis, transfer, and generation. In analysis, a source language parser is used to produce the syntactic representation of the source text. In the transfer phase, an equivalent intermediate representation in the target language is produced. In the generation stage, target language output is generated from the intermediate target representation. The output of this approach is better than that of DMT. Yet, the implementation complexity is very high. Each of the three-step in translation requires detailed consideration. However, the analysis and transfer steps are reusable components when multiple translation pairs are considered.

2.2.1.3 Interlingual Machine Translation Approach

In Interlingual Machine Translation, a two-step approach is used for translation. In the first step, the source text is analyzed and transformed into an interlingual language (language independent) representation. Then text in the target language is generated using this representation. The key advantage of this methodology is, the intermediate state is language independent. This is advantageous in multilingual translation systems as one step out of two can be generalized. Yet, the only interlingual system that was operational was KANT [9] system, that was used to translate Caterpillar manuals. Less popularity of this system is owing to the high complexity of the intermediate representation and this intermediate representation fails to represent the semantics.

2.2.1.4 Challenges in Rule-Based Systems

Rule-based systems were the initial implementations of MT. The implementations were highly dependent on language and detail analysis of both the languages was required. This required the involvement of skillful linguists in implementation. Also, the implementation details were very specific to the language pair as well as the direction. Therefore, implementation for a new pair or new direction required an additional amount of human work in setting rules [10].

2.2.2 Corpus-Based Machine Translation Approach

The main difference in corpus-based approaches from rule-based is, in the implementation process less or no human involvement is used in building rules. Instead, translated content such as a parallel corpus is given to the system to study the rules from the content and context. This approach requires a large amount of raw data as a parallel corpus to acquire the translation knowledge. The corpus-based approach is further classified into as Example-Based Machine Translation, Statistical Machine Translation, and Neural Machine Translation.

2.2.2.1 Example-Based Machine Translation (EBMT)

Example-Based Machine Translation (EBMT) approach is motivated by repeated translation work, where same text with minor variations (e.g., Only the proper nouns vary in the sentence) needs to be translated several times. The system tries to find the matching sentences/example sentences/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. Four steps are involved in this: example acquisition, example base management, example application, and synthesis [11]. This approach requires high quality aligned data. This is ideal for narrow, domain-specific translations where a considerable amount of repetition is found in the content to be translated.

2.2.2.2 Statistical Machine Translation (SMT)

Statistical Machine Translation (SMT) is a corpus-based machine translation approach, which is based on the statistical models that are built by analyzing a parallel corpus (Collection of text placed alongside with its translation) and a monolingual corpus. The original idea of SMT was initiated by Brown et al. [12] based on the Bayes Theorem, as statistical models worked successfully for speech recognition. Basically, two probabilistic models are being used, Translation Model (TM) and Language Model (LM).

Translation model is built using the parallel corpora, and it is used to identify the translation options of a source word/phrase. Language model is built based on the target side monolingual data and this is used in verifying the fluency of the translation options provided by the TM. The output is generated by maximizing the conditional probability of the target given the source language. This output generation process is called ‘decoding’ (discussed further in section 2.3.6). With the advancement, currently, SMT systems operate based on a log-linear model (Further detail on log-linear model can be found in [13]), with consideration to phrases, not words, as the translating unit. Further insight on SMT is discussed later in section 2.3 of this chapter.

2.2.2.3 Neural Machine Translation (NMT)

Recently, neural approaches based on deep learning techniques for machine translation had shown promising results for many language pairs over the statistical machine translation. This methodology does the full translation process with a single neural model. The commonly used approach is ‘encoder-decoder’ framework [14]. Here the source sentence is encoded into a vector, which is called the context vector. Then in the decoding process, the translation is generated for this vector. Since the translation process happens to the whole sentence in one step rather than in segments, the fluency of the output has been better than that of SMT approach [15].

Yet, the main shortcoming of NMT is that it falls back for the unknown words [16]. NMT systems do not work well for low-resourced setups, while SMT worked better than NMT in such cases. Tennage et al. [17] reported this behaviour for the Sinhala-Tamil pair as well.

2.2.3 Hybrid Approach

Hybrid machine translation is where multiple translation methodologies are used to build a single translation workflow, with the intention to utilize the advantages of different approaches. These hybrid MT systems have shown improvement in translation output [2, 18]. Different implementation variations have resulted in improvements, such as the following:

- Utilizing a corpus to build the RBMT system.
  Examples:
  - Use of parallel corpus to build a dictionary for an RBMT system [19].
  - Improving RBMT translation output by incorporating a language model [20].
  - Statistical post editing over RBMT output [21].

- Use of rules in corpus-based translation.
  Examples:
  - Pre-process the SMT source text in such a way, where the source sentence order matches that of the target sentence [22].
  - Post-processing rules over SMT output for a morphological generation. [23].
  - Use of rules to improve the statistical word alignment [24].
- Hybridizing between corpus-based approaches.
  Examples:
  - Use of neural-based (language) models in SMT system [14].
  - Incorporation of SMT model in NMT system. NMT does the prediction by taking consideration of the SMT recommendation on word selection [15].

2.3 Statistical Machine Translation

SMT treats the translation problem as a machine learning problem [25]. The system tries to learn how to translate by means of creating statistical models with the human translated text. The translation output is determined by identifying the output that maximizes an objective function [12]. The initial SMT system concept was built upon the noisy-channel model where the system tries to maximize the function 2.1, where t_{best} is the best translation that has the maximum probability based on the noisy-channel model:

$$\begin{aligned} t_{best} &= argmax_t(p(s|t) * p(t)) \\ s &= source\ language \\ t &= target\ language \end{aligned} \tag{2.1}$$

Where *p(s/t)* component is derived from the translation model, which is learned from the parallel data. *p(t)* is derived from the language model, which is derived from the target side monolingual data. Based on the span of text considered at the time of translation task (in translation model), the SMT systems are categorized as word-based SMT and phrase-based SMT systems.

Yet, this model, based on noisy-channel, allowed to have the influence of only two features, and the impact of each was equal over selecting the candidate.

Yet, besides the knowledge on translation equivalent and word combination, the translation process requires more information such as reordering and distortion in different degrees. However, the noisy-channel based model does not have the facility to cater to multiple models. Therefore, as an improved SMT version, log-linear model based SMT is used over noisy-channel model [26] where the objective function is as of function 2.2 (Since the model tries to maximize the summation of logarithm, it leads to a linear function. Refer Knoke et al. [13] for further mathematical insight).

$$\begin{gathered} t_{best} = argmax_t(exp(\Sigma\lambda_i h_i)) \\ h_i - model \ / \ feature \\ \lambda_i - model \ / \ feature \ weights \end{gathered} \tag{2.2}$$

This allowed a very rich set of features to be used. This facilitates to have multiple models and multiple sub-models under each model. Each model can have a different weight based on its influence over the translation task. The weight of each model was adjusted at the time of tuning (For further details, refer section 2.3.7) with reference to the set aside tuning set. The following are the common set of features used in the log-linear based model, where the first two are the sub-features that come under the translation model [27]:

- Bidirectional (i.e. Source to target and target to source) phrase translation probabilities.
- Bidirectional lexical probabilities.

- Language model probability.

- Phrase reordering / distortion model.

- Word / phrase penalty.

2.3.1 Translation Model

Translation model (TM) is used to calculate the likelihood of a target segment being the translation for a given source language segment. The model is learned from the statistics gathered from the parallel corpus (sentence-aligned parallel corpus). Based on the unit of text considered at the time of decoding (translation), the system is known as word-based SMT or phrase-based SMT.

2.3.1.1 Word-Based SMT

Word-based SMT methodology is where the unit of consideration for translation at a time is a single word. In other words, the translation model translates one word at a time. Though this has been the early version of SMT and no more the state-of-art SMT methodology, it paved the path for most of the current SMT approaches. There are five different word-based models which were proposed by Brown et al. [12]. They are known as IBM models as they were the result of the research at the IBM Watson research center. These different methods used different techniques to calculate the translation probability *p(s/t)*.

Generally, the translation model cannot be derived directly from the parallel data alone due to the sparseness in the data. Therefore, the given data is decomposed into a sequence of words. Then the required statistics are derived to estimate the probability distribution. IBM models suggest different algorithms to derive the statistics of a target word being the translation of a given source word. By utilizing this statistic information, the words in source sentences are aligned to the words in target sentences pair.

Figure 2.2 depicts a possible word alignment between an example Sinhala-Tamil sentence pair (translation direction Tamil- Sinhala; alignment direction Sinhala-Tamil).

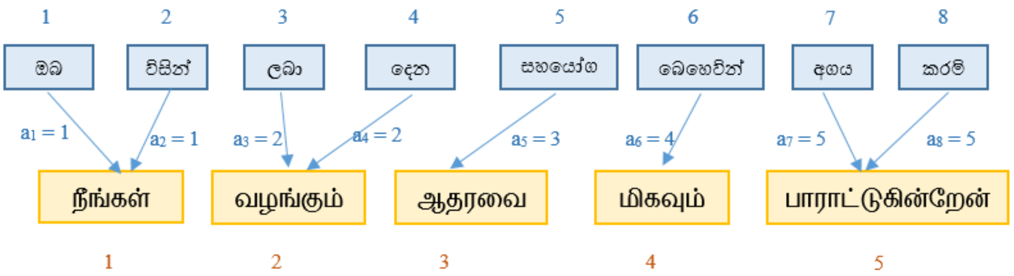


FIGURE 2.2: Sample word alignment between Sinhala and Tamil

The hidden alignment variable a_i that captures word-level correspondence between source(s) and target(t) are used to define IBM models. The conditional probability *p(t|s)* is expressed as a sum of the probabilities of hidden alignments between s and t as the function 2.3 [27]:

$$\begin{gathered} P(t|s) = \Sigma_a P(t,a|s) \\ a - vector\ of\ alignment\ positions\ a_i\ for\ each\ word\ t_i\ in\ t \end{gathered} \tag{2.3}$$

The *alignment function* is a mapping of each target (Sinhala) word position at position *i* to a source (Tamil) word position *j* as 2.4.

$$a : i \rightarrow j \tag{2.4}$$

Therefore, this mapping consists of many-to-one mappings and will not consist of one-to-many or many-to-many mappings.

There are instances where there is a word in the target sentence that does not have a corresponding word in the source sentence. In such cases, the alignment model tends to drop such words. To overcome this concern, a *NULL* token is added to the source sentence, and these isolated words are aligned to it.

As alignment is hidden and unknown, the lexical probability becomes an incomplete data problem. Therefore, as in machine learning, the incomplete data problem is addressed by using the Expectation Maximization (EM) algorithm [28]. EM is an iterative algorithm in which alternating steps fill the gap in the data and then train the model. In the initial step, a uniform lexical probability is given to each possible alignment. In the following iterations, based on the co-occurrence count of word pairs, better lexical probabilities are learnt. This iterative process is repeated until it is converged to good lexical probabilities. In each iteration, perplexity (A probabilistic measure of how well the model predicts a sample. The measure is used to compare models, the lower perplexity indicates that the model is good in predicting sample. Refer Jelinek et al. [29]) is used to evaluate and determine the convergence of the EM algorithm.

Brown et al. [12] proposed five different generative models (breaking up the process of generating data into smaller steps, modeling the smaller steps using probability distribution and combining them back) based on this word alignment theory. They were named from IBM model 1 until IBM model 5. Each succeeding model had an improvement over its predecessor by including extra information to the alignment function.

The EM algorithm is used at the time of training, to estimate the hidden parameters by maximizing the likelihood probability of the parallel training corpus. GIZA++ [30] and MGIZA++ [31] are two of the widely used toolkits that implement IBM models and Hidden Markov Models (HMM).

In IBM model 1 and 2, the models estimate the alignment using the lexical probability. They consider the source sentence to calculate the lexical translation probabilities. IBM models 3, 4 and 5 focus on the target sentence. In summary, key advances of the five IBM models are:

IBM model 1: lexical translation

IBM model 2: adds absolute alignment model (position of the aligned word)

IBM model 3: adds fertility model (deals with dropping a word or providing multiple words for a given word)

IBM model 4: adds relative alignment model (local alignment within a phrase)

IBM model 5: fixes deficiency (avoid placement of output words to the positions that have already been filled)

These IBM models do have drawbacks. The crucial drawback is that these models can align each target word to one source word only. However, in practical scenarios, many-to-many alignment is required, especially to translate expressions and idioms. Also, these models do not consider any contextual information in estimating translation probabilities. As an enhancement over these limitations, phrase-based models were suggested [32]. The following section briefly discusses phrase-based translation models.

2.3.1.2 Phrase-Based SMT

In phrase-based models, longer translation units (more than a single word) are considered to be the atomic units. Therefore more contextual information is captured by the translation. This leads to better translation than in word-based models. These multiword translation units are called phrases. Yet, there is no linguistic motivation in this phrase partition. In order to cater to this, it is required to generate a *phrase table* with phrases as entries. Apart from this, phrase-based models have the ability to handle simple reordering techniques as it considers a phrase as a single unit.

IBM models still play a vital role in phrase-based models as they are used in generating word alignment, which is an important step in training phrase-based models.

In word-based models, the alignment technique used is known as *asymmetric alignment*, where each target word is aligned to only one source word (one-to-many). In phrase-based models, the method used to make many-to-many mapping is termed *symmetrizing*. To create symmetrization, word alignment is trained in either direction (source-to-target and target-to-source) separately. This will result in two alignment matrices.

This word alignment information is used to extract phrase pairs. Phrase extraction includes following steps [26]:

- Performing asymmetric alignment of the parallel corpus in both (source to target and target to source) directions.
- Getting a high-precision alignment and a high-recall alignment by using the intersection and the union of both alignments, respectively.
- Start with the high-precision alignment points and add additional alignment points using heuristics.
- Looping over all possible phrases of the target sentence and finding the minimal source phrases that match each of them, to extract phrases.

Currently, phrase-based SMT systems are considered as the state-of-art SMT methodology.

2.3.2 Language Model

Language model is a crucial element in natural language processing. It models the fluency of the proposed target sentence with greater probabilities being appointed to sentences that are more practical in natural language [33]. A language model assigns a probability for each sentence that indicates how likely the sentence is to occur in the text (the target side monolingual corpus). The higher this probability, the sentence is considered better in the fluency. To reduce the complexity of the length of sentences, n-gram language models are deployed [34]. These n-gram language models use Markov assumption to break the probability of a sentence into the product of the probability of each word, given the history of previous words (the number of previous words considered depends on the order of n-gram).

The language model influences the word choice, reordering and other decisions that lead to language fluency. However, the fundamental challenge in this is handling data sparseness. This happens since the natural languages are widely diverse, and covering all the possible contexts of occurrences for a word in a corpus is practically infeasible.

Different smoothing techniques such as add-one smoothing [35], deleted estimation [35] or Good–Turing smoothing [36] are used to get a small probability from the available occurrences and to assign them to unseen occurrences. This helps to overcome the issue of assigning zero probability for unseen occurrences [37].

Apart from these smoothing techniques, interpolation [38] and back-off [35] techniques are used to address the problem of data sparseness. In interpolation, n-gram models of different orders are combined. In back-off models, in the initial step, the highest order model is consulted. If the occurrence is not found, it moves down to the lower order models. Various techniques are used to determine the back-off cost and to move to the elementary order models. One of the well-known technique is Kneser–Ney smoothing [37].

For languages with larger amounts of data (target side monolingual data), the language models lead to better results [2]. Yet extra measures are required to handle such large models as it is computationally challenging. Models are trained in the disk. However, since the model is needed to be loaded to the RAM at the time of decoding, efficient data structures and clusters are being used.

Two different methods are used to measure the quality of language models. The first way is an end-to-end evaluation. Here, the language model performance is evaluated by incorporating it into the relevant framework (in this case MT). Though this evaluation is the best, it is very expensive due to the high time consumption. The second approach is to calculate an independent language model quality measure based on a development set. The standard metric used for this is perplexity (PP) [29]. It measures how much probability is given to a set of actual (target) text.

Apart from the conventional statistical language models, neural network-based language models have also been used in recent research [39]. This is also known as continuous space language model or CSLM. It tries to overcome the disadvantages of back-off n-gram language models. One of the disadvantages with the statistical models is that the probabilities are estimated in a discrete space. This will not support the estimation of non-observed n-grams in the training data directly. However, in neural network-based language models, the words are projected onto a continuous space during training where a multi-layer model, jointly learns the word projection and the probability estimation [39].

2.3.3 Reordering Model

Reordering model is in-charge of the reordering of output words/phrases but at a cost. Two types of reordering models are being used in SMT. They are the distance-based model and the lexicalized model [2].

In distance-based models, the cost is computed by measuring the number of words that are skipped when phrases in the source text are picked in the same order of the target text. When the parameter is set closer to 0, monotonous translation is expected.

Since the distance-based model is not strong enough to handle reordering in specific phrases, lexicalized reordering model is introduced [40]. This model is learnt from the alignments obtained during the extraction of parallel phrases and is used to score the order in which the aligned words appear in the target text.

2.3.4 Word Penalty

Word penalty ensures that the translations are neither too short nor too long [41]. This model counts the number of words in the reference translation (with respect to the source sentence) and the weight of the model is adjusted at the time of tuning. If ω is the weight of the word penalty model when $\omega < 1$, the output is biased to be shorter than the source and when $\omega > 1$, the output tends to be longer than the source sentences.

2.3.5 Phrase Penalty

To generate translation hypotheses, the sentences are segmented into phrases. These can be either short or long. The required phrase length depends on language as well as the context [2]. Similar to the word penalty model, the phrase penalty model also affects the phrases in the output sentences, whether longer fewer phrases or shorter more phrases. Based on the adjusted weights, if ρ is the weight of the phrase penalty model when $\rho < 1$, the system will prefer longer phrases while $\rho > 1$ means the system prefers shorter ones.

2.3.6 Decoding (Search)

Decoding is a process of finding the best target sentence that maximizes the conditional probability *P(t|s)* based on the log-linear function. First, the decoder searches over for all possible translation options. Next, it scores the translation options using the log-linear based function. Then the translation with the best score is selected as the candidate translation.

However, there is an exponential number of hypotheses for each sentence. This makes the decoding problem NP-complete [42], which means the process of searching all possible translations, scoring them and choosing the best one is computationally too expensive even for a sentence of short length. Therefore, to reduce the number of translations generated, heuristic methods are applied. Yet, the heuristic might not find the best translation, but one close to it.

There are two reasons for not finding the best possible translation [2]. One is search error and the other is model error. Search error occurs when the SMT system fails to find the best translation in the search space. The model error occurs when the translation with the highest probability according to the models is not a good translation. This happens due to the deficiency in the training data.

Decoding for word-based SMT has comparatively higher complexity than phrase-based SMT, as the former needs to focus on word level reordering. Optimal A* search [43], integer programming [44], or greedy search algorithms [45] are being used for implementing the decoding algorithm for word-based SMT, while beam search stack decoding is widely used in phrase-based SMT [2].

In beam search algorithms, the decoder looks into all possible translation options in the phrase table as shown in Figure 2.3.

Decoding process starts with an initial empty hypothesis. Then translation hypotheses are constructed from left to right. The hypotheses are expanded by considering the available translation options (refer Figure 2.4). Along with that, the source translation vector is updated. The translation probability for each of them is calculated.

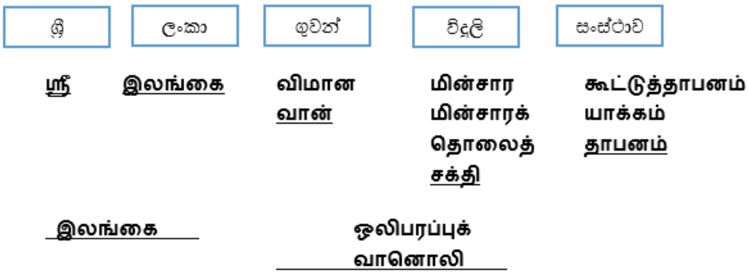


Figure 2.3: Possible Tamil translation options for a sample Sinhala sentence

To limit the exponential growth of this search space, different techniques are being used. Hypotheses recombination (i.e. Combine similar hypotheses that cover the same source translation, but have different scores), and pruning out bad hypotheses with worse scores from the hypotheses stack is used to reduce the growth of the search tree. In order to prevent pruning out good future hypotheses, future costs of hypotheses are estimated at each step. Until all the source words are covered, all of the hypotheses are continued.

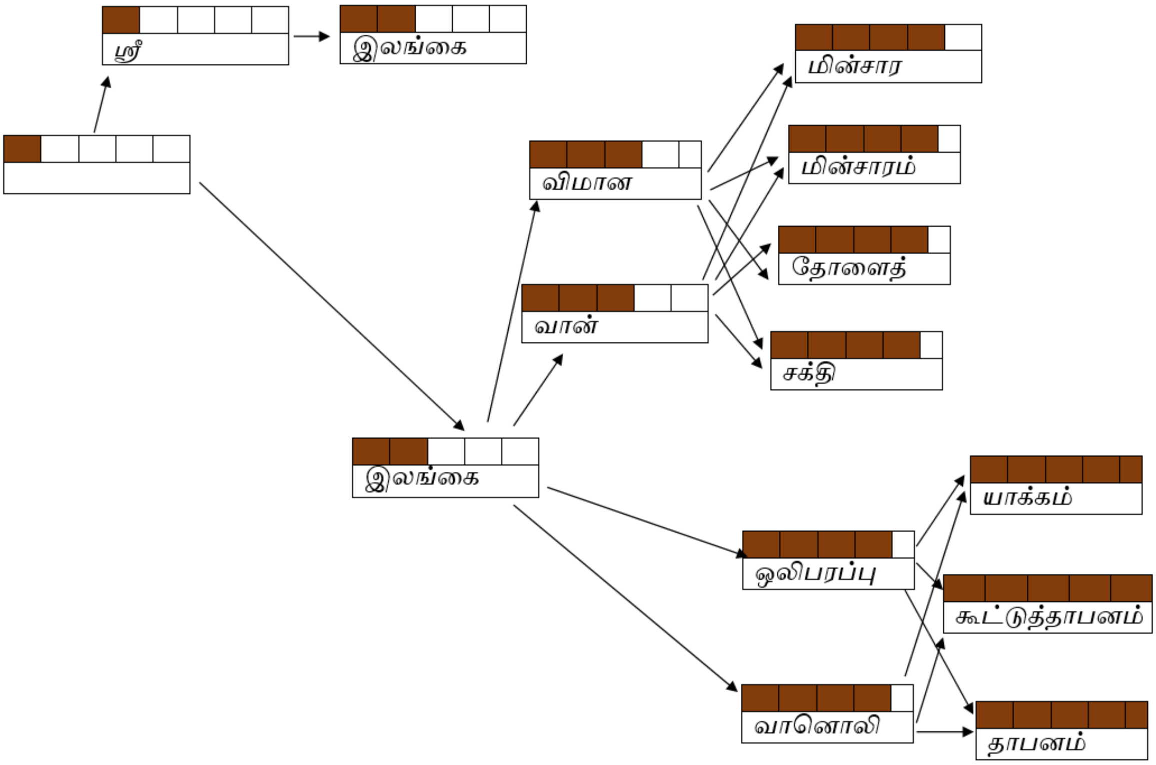


FIGURE 2.4: Sample diagram of the decoding process

In case where there are multiple completed hypotheses found, the hypothesis with the highest probability is selected as the best translation.

2.3.7 Tuning

Log-linear models allow multiple features with different weights, while noisy-channel models allow only two features with the same weight. Generally, in log-linear models, the weights of the features are adjusted using a supervised algorithm. This algorithm is used to maximize the translation quality on a held-out parallel data set (tuning data). The quality is measured by an automatic evaluation metric. This process is called tuning. The default algorithm used for this process in machine translation is Minimum Error Rate Training (MERT) [46]. Since MERT does not scale well to a large number of features [47], other tuning algorithms such as Margin-Infused Relaxed Algorithm (MIRA) [48, 49], and the Pairwise Ranked Optimization (PRO) [50] are also used.

The process of weight adjustment for the features is as follows:

- Initially, random weights based on some heuristics are set for each feature (for each feature h_i weight is set as λ_i.)
- N-best translations of the development set (tuning set) are retrieved with current weights (λ_i).
- Compare the objective score (using an automatic evaluation metric) of the n-best translation with the previous run.
- Re-estimate the weights (λ_i).
- Iterate until weights have converged.

2.3.8 Evaluation

In order to quantitatively measure the quality of the output of machine translation, it should be evaluated. This evaluation can be done either using manual evaluation (by human) or automatic evaluation (using computers). Each method has its own pros and cons.

Manual evaluation is based on two human judgment factors, adequacy, and fluency [51]. Adequacy is the measure of how much of information is contained in the output. Fluency is the measure of how fluent the output language flow is. The scores are given in the range of 1-5, where 5 is a perfect translation and 1 is an incompetent translation.

Since manual evaluation is costly with regard to time and money, automatic evaluation is widely utilized in most of the machine translation evaluations.

In an automatic evaluation, the evaluations are based on some correlation metric between the translation output and one or more reference translations which are translated by humans. This evaluation technique is much useful in evaluating relative translation quality between different experiments. Though the automatic evaluation is widely used in the evaluations, still it is an open field for research [52].

The most established automatic evaluation metric in the field is Bilingual Evaluation Under Study (BLEU) [53]. The evaluation happens by measuring n-gram occurrences between a given translation and the set of reference translations and then calculating the weighted geometric mean. This metric is precision based metric where the score is a fraction of n-gram matches to the n-grams available in the reference output.

Evaluation metric by National Institute of Standards and Technology which is known as NIST [52] is another evaluation metric used in machine translation evaluation. This evaluates in the same manner as of BLEU, though more weight is given for the correct n-grams which are rare.

Translation Edit Rate (TER) [54] is an automated translation evaluation metric, which is based on the minimum number of edits required to match the reference. Edits include insertion, deletion, substitution, and shift of words.

Human-targeted Translation Edit Rate (HTER) [54] is an extended version of TER. Here the human effort is involved over the machine translated output to edit to get a fluent translation out of it. This edited output is used as the reference for the evaluation.

Word Error Rate (WER) [46] is an evaluation metric based on the edit distance derived from the Levenshtein distance (a similarity measure between two strings where the distance is the number of deletions, insertions, or substitutions required to transform the source into the target).

Metric for Evaluation of Translation with Explicit Ordering (MTEOR) [55] is an evaluation metric that is based on the harmonic mean of uni-gram precision and recall, with recall weighted higher than the precision. The accuracy of this metric is higher than that of BLEU [53]. Yet, this evaluation procedure is more complex as it requires tuning.

The automatic evaluations are based on the comparison of the output to the reference translation. The system may penalize at times for synonyms and different writing styles in cases where both, the translation and reference are correct.

All the automatic evaluations depend on either the edit distance or n-gram similarity. Yet, it fails to analyze the semantics of the output vs the reference. However, the automatic evaluations are widely used in MT evaluation task due to its time and cost efficiency [2]. Out of the various automatic evaluation methodologies available, BLEU is widely used in research [2] due to its best compromisation between the quality and effort.

2.3.9 Common Challenges in SMT

SMT systems are built upon specific parallel corpus and monolingual corpus, hence the systems are closed vocabulary. Due to this, SMT systems face following set of challenges.

- **Out-Of-Vocabulary (OOV)**: Some words in the source sentences are left untranslated by the MT system since it is unknown to the translation model. The OOV words can be categorized as named entities, inflection forms of verbs and nouns, numerals and dates.
- **Unknown target word/word combination to the language model**: When the word or sequence is not known to the language model, the system suffers from producing fluent output as it does not have sufficient statistic on selecting among the word choices.
- **Mismatch between the domain of the training data (parallel and monolingual corpus) and the domain of interest**: Writing style and word usage can have drastic differences from domain to domain. For example, the writing style of official letters differs much from that of news articles. The meaning of words may vary depending on the context or domain. For example, the word ‘tablet’

is translated to a ‘pill’ if the considered domain is medical, while to ‘ hand-held computer’ if the domain is computing.

- **Ambiguity**: Ambiguity is the problem of a word having multiple meanings or can be understood in different ways. As in many other NLP tasks, even in MT, ambiguity is a major challenge. The ambiguity could be lexical, syntactic, semantic, or pragmatic.

- **Multi-word expressions such as collocations and idioms**: The translation of multi-word expressions is beyond the level of words. Therefore, in most cases, they are incorrectly translated.

- **Mismatches in the degree of inflection 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.

- **Different word order patterns**: Different languages do have different word ordering (some do have subject-object-verb while some others have subject-verb-object). When translating, extra caution is needed to make sure that the output flow is fluent.

2.3.10 Prior work in Sinhala-Tamil Translation

The Sinhala language is a branch of Indo-Aryan language family, while the Tamil language is a branch of Dravidian language family [56]. Apart this, the syntactic divergence between the two languages is lesser when they are compared with English [56]. For instance, both the languages are composed of subject–object–verb sentence structure. Although, either of the languages are highly inflected and morphologically rich. However, when considering the languages, Sinhala and Tamil, the amount of research that has been carried out in translation so far is very little. One reason could be the lack of freely available data and linguistic resources. The published work is, in general, experiments on the feasibility of translation in the open domain. The data sources are news articles with marginal amounts of parliament order papers.

As the initial attempt, as a study on the feasibility of SMT, for the pair Sinhala and Tamil was done by Weerasinghe [56]. ISI-Rewrite decoder was used for the translations while CMU-Cambridge Statistical Language Modeling Toolkit for language model building and Giza++ for translation model building. A parallel corpus of 5000+ parallel sentences was used for this experiment which were Sri Lankan politics and culture related news articles. The evaluation scores were very low. Yet, the comparison was done with their previous Sinhala-English system. The evaluation scores were higher than that of Sinhala-English system where they concluded that this is due to the more linguistic similarity between Sinhala-Tamil than that of Sinhala-English.

With a parallel corpus of 5000+ parallel sentences from parliament order papers, a system was analyzed by Sripirakas et al. [57]. Their system used Moses as the translation engine with Giza++ for Translation model building and SRILM for Language model building. The experiments assessed the impact of tuning over the SMT model and concluded that the Tamil-Sinhala direction produced better translation than in the Sinhala-Tamil direction.

Another significant experiment was carried out by Pushpananda et al. [58] where they examined the performance of SMT systems against the volume of parallel data. A parallel corpus of 25,000 lines from the open domain, Sinhala monolingual corpus of 850K sentence, and Tamil monolingual corpus of 407K were used to evaluate the system. Moses along with Giza++ and SRILM were used in experiments. The results concluded that though adding more data showed improvement, still, the system requires more data to perform well. They further state that morphological richness of both the languages is one of the reasons behind the lower scores.

Rajpirathap et al. [59] did a study on the system behaviour in absence and presence of tuning step for the weights of different models and features (language model, translation model, word alignment, lexical reordering). The source was 5000+ parallel sentences from parliament order papers. Here, the focus was on developing a web-based translation system.

Extending their initial work [58], Pushpananda et al. [4] used the same data set in another study on improving the existing system with morphological analysis. Here, Mofessor algorithm [60] was used to incorporate an unsupervised morphological analyzer to the system. The system was trained on morpheme-like units rather than the surface forms. The results reveal that the system significantly reduces the number of OOV words when this unsupervised morphological analyzer is incorporated. The experiments were carried out only in Tamil-Sinhala direction. Translation service based on this research is currently available under the name 'Subasa' at http://translate.subasa.lk/si2ta.php.

Apart from these published work, Google provides online translation form Sinhala-Tamil and Tamil-Sinhala in Google Translate[3].

2.3.11 Challenges in Sinhala-Tamil Machine Translation

Apart from the common SMT related challenges, Sinhala-Tamil do have their own set of challenges. Few key points are listed below [4, 56–59]:

- **Low resource:** Sinhala-Tamil pair lacks freely available parallel data as well as decently performing linguistic resources and tools. This causes the MT research on this pair a practically hard problem.
- **Both are highly inflectional:** Sinhala, as well as Tamil, are highly inflected languages, and the degree of inflection is different.
  For example, the words 'ගෙදර' /gedara/ and '**வீடு**' /weedu/ mean 'home' in Sinhala and Tamil respectively.
  Following are the corresponding equivalents for 'to home', 'from home', and 'at home' which are the inflectional variations of the base word.

[3] translate.google.com/

‘ගෙදරට’ /gedarata/, ‘ගෙදරින් ’ /gedarin/ ‘ගෙදරේ’ /gedare/

‘**வீட்டிற்கு**’ /weettitku/ ‘**வீட்டிலிருந்து**’ /weettilirundu/, ‘**வீட்டில்**’ /weettil/.

However, the word ‘a home’ is an inflectional variation in Sinhala (‘ගෙදරක්’ /gedarak/) while it is represented with an article in Tamil as ‘**ஒரு வீடு**’, /oru weedu/.

- **One-many style:** Apart from the mismatches in the inflection, there are conflicts in word alignment in Sinhala and Tamil. Sinhala has many compound words that are represented by a single word in Tamil. Each word in such compound Sinhala word has a corresponding Tamil translation. This makes the word alignment task tedious.

- **Abbreviations and initials:** When it comes to the initials as well as abbreviations, in either language, there is no norm followed. In some cases, the transliteration of English letters is used while in some cases the first letters in the language of consideration are used. Not having the knowledge on how the abbreviations were derived makes translation confusing. Also, generating the English phonemes by referring the Sinhala or Tamil script is also confusing as there are mismatches in phonemes between the languages. For example, there are no different phonemes in Tamil to distinguish the English letter ‘B’ and ‘P’.

- **Orthographic error:** As both, the languages consist of larger alphabet sizes than English, the keyboard systems for Sinhala and Tamil are more complex than that of English. In practical use most of the time, non-Unicode fonts are used in document processing, sometimes with local customization over the font. Though from the point of human reading this creates 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 orthographic errors in the data.

2.4 Techniques for Improving Translation Quality

As the main focus of this thesis is to apply different pre-processing techniques to improve the translation quality for a domain-adapted SMT with limited resources, this section describes on related work on such techniques that were explored in our work. Areas covered are domain adaptation, terminology integration, and out-of-vocabulary handling.

2.4.1 Domain Adaptation

It is well-known that the performance of SMT degrades when the test data set highly deviates from the training data set. The reason is that the underlying statistical models try to approximate the empirical distributions of the training data, which always represent the characteristics of the training data. The languages are highly variable with respect to several dimensions, such as style, genre, domain, topics, etc. SMT systems developed for open-domain are not capable of addressing these domain-specific variations, as they are trained using general data. The best way to set up a domain-specific SMT system is to build an SMT system solely with a large amount of in-domain data. Yet, finding such a large amount of in-domain data is practically infeasible for many languages.

An amount of research has been done on domain adaptation, either with parallel or monolingual in-domain data (e.g: Koehn and Schroeder [61], Bertoldi and Federico [62], Hildebrand et al. [63]). Utilizing in-domain corpora along with out-domain corpora, with the same priority, is a challenge. If a single translation model is trained with the entire parallel data (in-domain and out-domain), the domain specific features get overwhelmed by the out-domain data [61]. Nonetheless, the SMT system may fail in generalizing general language characteristics, if only the in-domain corpora are used. This may lead towards low quality translation [64]. Therefore, domain-specific SMT engine should occupy the generalizations of an engine trained on large parallel corpora, while not failing to have domain relevance.

Two major approaches have been used in domain adaptation in previous research. These methodologies are applicable depending on the data ratio difference between the in-domain and out-domain data:

1. **Using an open-domain system to fine-tune into a specific domain**

   Koehn and Schroeder [61] suggest the use of cross-domain adaptation. A substantially smaller size in-domain data is being exploited over a substantially larger size out-domain data, using linear interpolation technique.

   Foster and Kuhn [65] used the concept of mixture modeling [66], to develop dynamic domain adaptation. For multiple different domains, adaptation was done using a cross-domain technique. By analyzing the input text, a mixture model is generated based on an unsupervised clustering method and mixture weights are estimated dynamically. This is an extended version of the system by Koehn and Schroeder [61], as they cater to domain adaptation for multiple domains within a system in a dynamic manner.

   Civera and Juan [67] used mixture modeling in domain adaptation, to enhance the word alignments by intervening the alignment process to generate topic-dependent word alignment over general alignment. Yet they doubt on the applicability of this technique, as the performance of SMT depends on many factors.

   The approach based on these adaptation techniques is feasible for a pair that already has a good open domain system (has reasonably enough open domain data) while having a minimal domain-specific data. Simply, the in-domain data is used to fine tune the open domain system to a specific domain.

2. **Data filtration techniques to extract data from the open-domain corpus that is similar to the in-domain data**

In order to guarantee that the data is from a same or similar domain, different filtration techniques are used in collecting and filtering open-domain monolingual data [68], as well as parallel data [63].

Data filtration is the process where a given set of data is being processed to remove the less similar sentences from an out-domain corpus with reference to the given in-domain corpus (tuning set).

One of the measures used for filtration is perplexity [2]. It is used to evaluate how similar a given sentence to a reference model is. Different techniques are being developed based on this concept to filter parallel as well as monolingual data. This concept is used in SMT domain adaptation with the motive to reduce the influence of highly deviating or less similar sentences.

Gao et al. [69] suggest the use of a simple perplexity metric of sentences based on the in-domain language model, to filter off the sentences that have a perplexity higher than a threshold. Moore and Lewis [70] convey the idea of using the cross-entropy difference between the in-domain and out-domain language models as the measure for filtration. They point out that this technique works better in reducing the perplexity than the method by Gao et al. [69]. Both these techniques can be applied for parallel as well as monolingual data. Axelrod et al. [71] suggest the use of bilingual cross-entropy differences, which can only be used over parallel data.

Though these techniques are based on minimizing the perplexity, improvement in the SMT translation quality is not assured [70, 71], as the behavior of SMT systems depends on multiple factors.

2.4.2 Terminology Integration

Terminologies are words or compound words that give specific meanings in specific contexts. This includes technical terms, named entities, designations, nominal phrases, and multi-word expressions. When it comes to professional translation, for the translation of such terminologies, more emphasis is given on the correctness and consistency of them based on the predefined reference/collection. Since terminology is a key element in government official documents, it is important that machine translation used in the translation workflow provides support to correctly handle terminology. Two key quality requirements need to be considered:

1. **Terminologies need to be translated correctly:** The translation needs to be as of in the provided reference term collection.
2. **Terminologies need to be translated consistently:** When multiple instances of the same word exist in a given document, the same translation should be given to all of those instances.

SMT systems face challenges in fulfilling the first requirement, since the system may not be able to identify sufficient context. In SMT systems the second requirement becomes more challenging than in rule-based systems since the system translates based on statistics rather than definite rules.

Therefore, explicit attention is required to translate terminology. This will result in an improvement of the overall translation.

The following are few translation issues in SMT when no explicit support is given for terminology integration.

- Terms may not exist in the SMT models: The term is an out-of-vocabulary and the term is left untranslated by the system.
- Terms may be translated into incorrect equivalents: A term may have multiple translation equivalents where not all the options are appropriate to the given context.

- Multi-word terms may be translated word by word: when non-breakable multi-word expressions are translated by splitting in between, it will result in word-by-word translation of words. This will lead to translation of words to their literal meaning, which leads to incorrect translations. For example, the translation for the word ‘ජංගම දුරකථන’ (/jangama durakathana/ (mobile phone) is ‘**கையடக்க தொலைபேசி**’ /kayyadakka tholaipesi/ (handheld phone) in Tamil. When this is translated word by word, though the literal meaning is correct, it will not be a fluent translation.

- Breaking of morpho-synthetic agreement between the constituents while translating : Morpho-synthetic agreements play a vital role in the translation of morphologically rich (highly inflected) languages such as Sinhala and Tamil. For example: for the term ‘බන්ධනාගාර අධිකාරී’ /bandanagara adikari/ (Prison Superintendent), the correct translation should be ‘**மறியற்சாலைக் கண்காணிப்பாளர்**’ Yet the word ’බන්ධනාගාර’ is an inflected form of the base word ‘බන්ධනාගාරය’ which has multiple forms in Tamil.

Terminology integration can be performed in two levels, in SMT systems [72]. 1) Statically (training the system with terminology) 2) Dynamically (when translating using a predefined SMT system). In static methodology, the knowledge of terminology influences the training procedure. In dynamic methodology, knowledge is influenced at the time of decoding with no changes in the existing models, by deploying pre, and post-processing techniques over the text to be translated. The figures 2.5, 2.6 depict the conceptual design of static and dynamic integration methodologies, respectively.

Many previous research work report improvement in translation quality for static as well as dynamic integration of terminology in SMT systems.

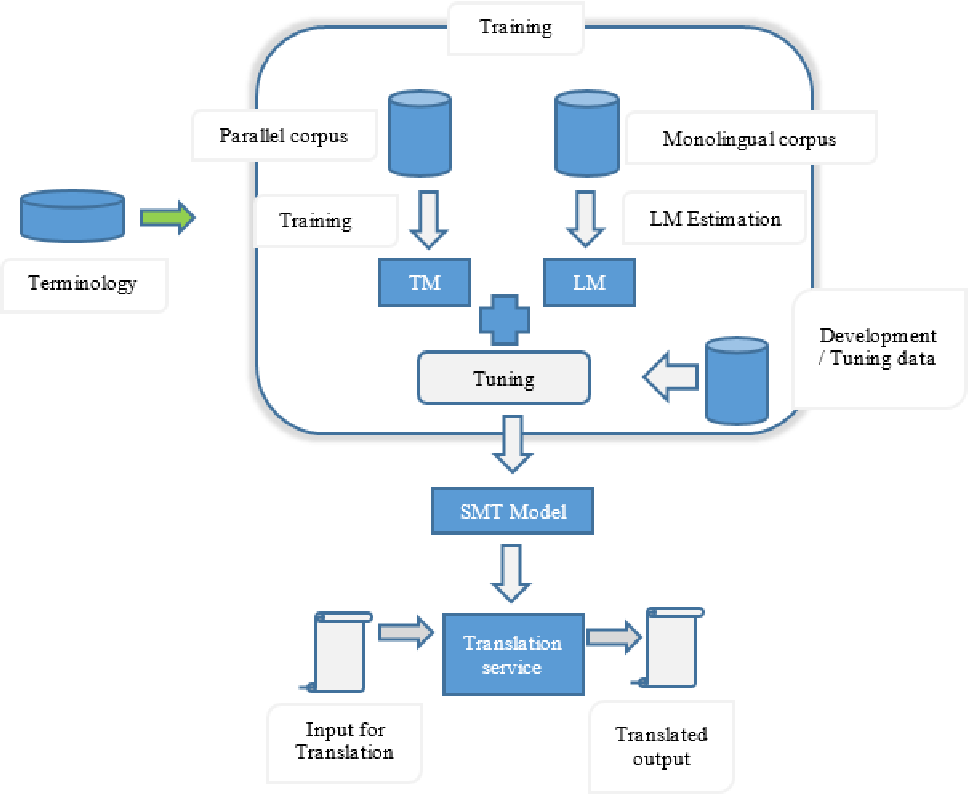


Figure 2.5: Conceptual design of static integration of terminology

2.4.2.1 Static Integration

Terminology integration is indirectly addressed in past research under multi-word expression (MWE) integration to SMT.

Initial work was done by Ren et al. [73] for Chinese-English pair. The experiments were carried out for medical domain (60k - parallel lines of sentences), and chemical domain (80k - parallel lines of sentences). The MWEs were extracted from the parallel data itself. Three different methodologies were experimented. They are:

1. Re-training the system with the list added to the parallel corpus (since the list is from the same corpus, it puts more emphasis on the expression).

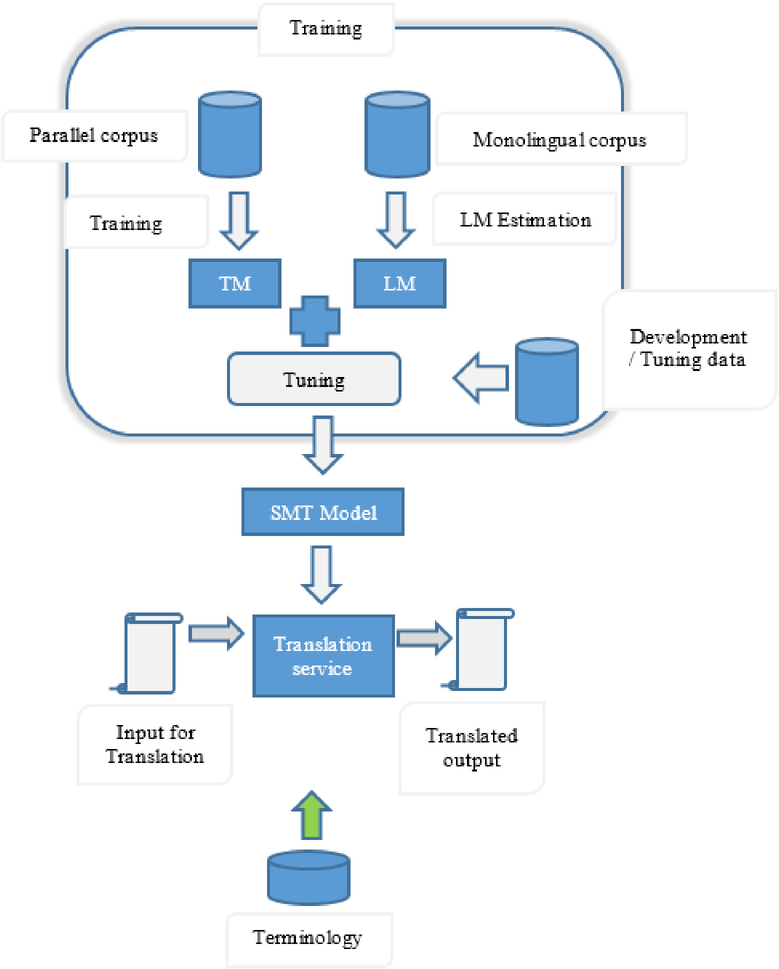


FIGURE 2.6: Conceptual design of dynamic integration of terminology

2. An additional new feature is added to the phrase table. For the phrase entries that have the correct MWE translation the value is set 1, otherwise, it is set to 0.

3. Use of multiple phrase tables. Apart from the phrase table generated from the parallel corpus, use the list to generate another phrase table. This phrase table is created with all the probabilities set to be 1.

Here results revealed that all the above techniques performed better than their baseline, while the technique with a new feature (second technique) outperformed.

Carpuat and Diab [74] experimented on the usefulness of the knowledge of monolingual MWEs on English-Arabic system with 2.5 M parallel sentences. The MWEs were extracts from WordNet and 500 high-frequency phrases from the phrase table. Two techniques were experimented.

1. The source MWEs were concatenated together with underscores as a single unit before the training process.
2. A new feature is added to the phrase table to indicate the number of MWEs present in the phrase.

Out of the above two, considering the MWE as a single unit performed better.

Bouamor et al. [75] experimented three different techniques for English-French system with the bilingual/parallel list extracted from the parallel corpus (100k lines of parallel sentences) itself. The first technique was based on the retraining as of in Ren et al. [73]. In the second technique, the entries were added to the phrase table with the lexicon probabilities set to 1. In the third technique, an extra feature is added to the phrase table of the second method to indicate the presence of MWEs. Here the first method performed better while the third method under performed.

As an extension of Carpuat and Diab [74] on monolingual MWE integration, Ghoneim and Diab [76] did a study on different types of multi-word expressions (named entities, nouns, verbs, adjectives and adverbs). Experiments were carried out and evaluated separately for each type of MWE. Results revealed that different methodologies work better for different MWE categories.

Skadiņš et al. [77] experiment on the use of bilingual terminology lists to do domain adaptation over a system with a large amount of open-domain data (5,363k lines of parallel sentence, 33,270k lines of monolingual sentences) and very small amount of in-domain data (tuning set -1745 lines of parallel sentences, testing - 872 lines of parallel sentences), for English-Latvian. Three different term lists were used in evaluations. They were:

1. A list manually extracted from the in-domain parallel corpus

2. A list automatically extracted from a comparable corpus
3. Entries from a term bank, which is manually compiled by professional translators. Entries were in their base forms.

Two methodologies were experimented:

1. The lists were added to the parallel corpus for training, and the target side was added to the monolingual corpus for language model training.
2. A new feature (ternary) was added to the phrase table to indicate the presence of term and correct translation for the term (term aware phrase table).

The results revealed that the use of the extracted list performed better than term banks as the extracted ones had the inflection variant.

Arcan et al. [78] showed significant quality improvement for English-Italian pair based on the fill-up technique. The phrase table created using the list was used to fill the gaps in the primary phrase table. An extra new feature was added to the phrase table to indicate the origin of the entry. The list was extracted from a monolingual corpus of either language.

In all previous cases, the research has been carried out for language pairs with a reasonable amount of parallel data. In most of the cases, bilingual lists were extracted from the parallel data itself. Therefore, the experimented methodologies put more emphasis on the available data, while the presence of these terms in the parallel corpus helped in providing context for the terms.

2.4.2.2 Dynamic Integration

One of the common issues with terminology integration is that the term is not known to the translation model. When the model cannot be re-trained, the SMT system can be provided with runtime integration for existing terminology lists. A number of experiments have been carried out in past on integrating terminology to the SMT system without re-training the models. Carl and Langlais [79] showed that use of term dictionaries to pre-process the source text gave an improvement in translation quality of English-French system. For English-Russian pair, Babych and Hartley [80] showed that the inclusion of named entities into 'do-not-translate' list by pre-processing the source text and not translating them by SMT system, improved the translation quality.

Apart from this, based on the Moses SMT framework's support for data input in XML format, research has been carried out to provide external translation options for terms. Arcan et al. [78] used this technique to identify the exact matches and provided translation equivalents for English-Italian system. In case when multiple translation equivalents were found, source contexts with Wikipedia documents were used to perform context-based disambiguation.

Similar kind of approach was experimented by Pinnis [81] for many language pairs (English-Latvian, English-Lithuanian, English-German, and English-Estonian). The focus was given in exploring different methodologies for term identification and inflection form generation.

2.4.3 Handling Out-of-Vocabulary Words

SMT systems create their models by analyzing the training data provided. Translations are picked from the translation model that is generated from the word alignment of this (parallel) training data. The system finds it difficult to translate the words that are not present in the translation model. Such words are left untranslated or dropped in the translation output. These words are termed as unknown words or Out-of-Vocabulary (OOV).

When the SMT system is a low-resourced setup, the vocabulary coverage is less. Due to this high sparseness in the data, the suffering from OOV is higher. Apart from that, for the morphologically rich languages such as Tamil and Sinhala, as the SMT systems consider each inflection form as a different word, OOV becomes more challenging.

These OOVs can be categorized as named entities, inflection variations, technical terms, compounds, misspelled words or foreign words [82]. Based on the type of OOV addressed, different approaches are tried out. Integration of transliteration models, POS and morphology integration, and use of paraphrases are few [2].

Habash [83] for an Arabic-English system handles OOVs by augmenting the phrase table with new entries based on morphological analysis, transliteration, spelling correction, and dictionary lookup. Habash and Metsky [84] present another technique for Urdu-English, where the seen morphological variants are used to find possible translations for OOVs.

However, these methods are heavily dependent on language-specific resources and linguistic properties. Therefore, these techniques are not applicable for languages apart from the one on which they were experimented on, as well as for low-resourced pairs that lack reasonably performing linguistic tools (i.e. morphological analyzer).

Out of the above-mentioned approaches, paraphrasing has been used in many previous research. Paraphrases are the alternative way of expressing the same idea, in a different way, in the same language. Previous approaches differ in the ways how the paraphrases were found/generated.

A domain-specific OOV handling was experimented by Banerjee et al. [85] for technical forum data for an English-French setup. The OOV were classified based on their type (terminology, spelling errors, content words, URLs, email addresses, and fused words) and they were post-edited using supplementary parallel data and spell checker.

Callison-Burch et al. [86] use bilingual pivoting to generate paraphrases for the OOV words. Here another third language is used as an intermediate version to find translations for the words that did not find a direct translation between the source and target sentences. Experiments were carried out for Spanish-English and French-English setups, with the parallel data between the source language and Danish, Dutch, Finnish, French, German, Italian, Portuguese, and Swedish were used as the pivot language to generate the paraphrases. Yet, this is ideal for a language pair that has reasonably enough parallel data with another common language.

As a solution to this, Marton et al. [87] came up with an approach based on distributional semantic similarity measure over a source-language corpus for English-Chinese and Spanish-English. The original phrase table was augmented with new entries generated for the OOV based on paraphrasing. A new feature is included to the phrase table to differentiate the original entries from the augmented entries.

Razmara et al. [88] used a graph propagation technique to generate paraphrases for French-English setup. The graph is built up from source side monolingual data along with the source-side of the parallel data available. The nodes with the associated meanings were linked together, while target-side translations along with their feature values were annotated for the nodes with translations in the phrase-table. Then to propagate translations from labeled nodes to unlabeled nodes a graph propagation algorithm is used.

Chu and Kurohashi [89] experimented on the use of word-embedding [90], semantic lexicons (using Word-Net [91], FrameNet [92], and the Paraphrase Database (PPDB) [93]) and a combination of both as of Faruqui et al. [94], to generate paraphrases for the OOVs for English-Chinese setup. Here the combination of both the methods gave the highest scores.

2.5 Summary

In this chapter, the introduction to machine translation, its history, and different approaches are discussed. Since the thesis is based on statistical machine translation; more focus was given on the basics of SMT, its functional components, and evaluation methodologies. This is followed by the discussion on general challenges in SMT. As the language pair of consideration in this thesis is Sinhala and Tamil, summary on the previous work and challenges faced by this pair are discussed. Compared to the well-performing SMT systems, Sinhala-Tamil pair is still in the incubation stage and lacks reasonable amount of resources. This requires extra effort such as pre-processing to improve the system with available resources. Since this thesis focuses on such techniques, the latter part of this chapter focuses on a few techniques that were used in improving the translation quality of SMT in the past. The areas discussed are domain adaptation, terminology integration, and out-of-vocabulary handling.

Chapter 3

The Baseline Si-Ta SMT System

3.1 Introduction

Sinhala-Tamil translation plays a vital role in Sri Lanka as both are considered as the official languages of the country. However, the number of individuals having a decent knowledge in both the languages is very minimal. This requires to publish each of the government related information in either language to provide each citizen an equal right to access the information. This process requires translation of content before publishing (most of the time from Sinhala to Tamil as original documents are prepared in Sinhala). And the translation process is done manually. However, this process functions inefficiently due to lack of human translators. This further increases the requirement for finding an alternative option to speed up the translation process. As a solution, using machine translation output as an intermediate translation version for the translators to work-on by post-editing can be used to speed up the translation task [95].

With this focus, the Center for National Language Processing (CNLP) [1]of University of Moratuwa, and the Department of Official Languages, Sri Lanka joined hands to implement a machine translation system to assist the professional translators. The system was named ‘Si-Ta’.

This chapter discusses the groundwork on building the Si-Ta system. This includes data gathering, pre-processing, experimenting with available tools and techniques, and building the initial baseline system.

3.2 Selection of Translation Methodology

Out of the available methodologies (rule-based, example, statistical, neural), the statistical machine translation methodology was preferred due to the following reasons:

- Rule-based MT systems require strong linguistic tools (morphological analyzer, part-of-speech tagger etc.) and greater involvement of linguistics, as a detailed analysis of the languages is essential to define the rules [26]. Yet both the languages are low-resourced. Therefore, greater human involvement is required.
- Example-based MT systems are ideal for translation tasks with repeated similar kind of work (ideal for translations of user manuals) [96]. Yet when it comes to government official documents, variations are found from organization to organization.
- Though neural MT is said to be the best performing methodology as of today, it requires parallel data in larger quantities (minimum in hundred thousands) [97], which is not feasible for a pair of low-resourced languages.

3.3 Data Source Description

Sinhala-Tamil is a low-resourced language pair with a barely minimum freely available ready to use parallel corpus. Therefore, it was required to start the work from the initial stage of data gathering. For this, government official letters and government administration related documents were considered in the data preparation. Data from the following sources were gathered:

[1] https://www.mrt.ac.lk/web/nlp

- **Government official letters:** Consist of letters from the Department of Official Languages, divisional secretaries of various provinces, police stations, and universities. The text consisted of a higher number of short phrases as the letters consisted of sender/receiver names, address, designations, and salutations.

- **Government circulars:** This includes the circulars from the Department of Education, as they were available on the web.

- **Annual reports:** Annual reports from different government ministries. The documents had many entries such as lists and tabulated data.

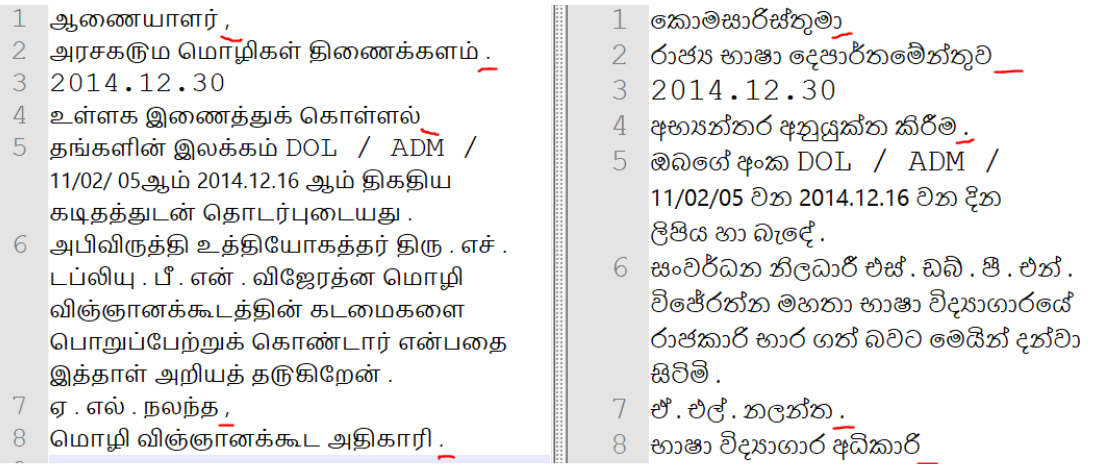
- **Parliament order papers:** Includes parliament proceedings. The text is in the form of question-answer format.

- **Establishment code**: Consists of the establishment code for government institutions. The document consists of technical term definitions and regulatory statements.

However, most of these documents were in a ready to be used state. Some were only in hard copy format in one language. The documents that were in soft-copy form were sometimes either in a non-Unicode form or were having locally customized fonts.

The single source documents were manually translated with the help of human translators. Documents in hard-copy form in both languages were digitized (typed into electronic form) by typists. For the other documents that were obtained in PDF format, a custom developed tool was used to extract data. Still, many font abnormalities were observed due to localized customization of the fonts. Those were manually corrected. Finally, using the semi-automatic sentence alignment tool created by Hameed et al. [98] was used create parallel data.

Table 3.1 summarizes the statistics of the data used for the parallel corpus. As there were many mismatches in the punctuation (refer sample Figure 3.1) these mismatches were eliminated using a script.



1 ஆணையாளர் ,
2 அரசகரும மொழிகள் திணைக்களம் .
3 2014.12.30
4 உள்ளக இணைத்துக் கொள்ளல்
5 தங்களின் இலக்கம் DOL / ADM / 11/02/ 05ஆம் 2014.12.16 ஆம் திகதிய கடிதத்துடன் தொடர்புடையது .
6 அபிவிருத்தி உத்தியோகத்தர் திரு . எச் . டப்லியு . பீ . என் . விஜேரத்ன மொழி விஞ்ஞானக்கூடத்தின் கடமைகளை பொறுப்பேற்றுக் கொண்டார் என்பதை இத்தாள் அறியத் தருகிறேன் .
7 ஏ . எல் . நலந்த ,
8 மொழி விஞ்ஞானக்கூட அதிகாரி .

1 කොමසාරිස්තුමා
2 රාජ්‍ය භාෂා දෙපාර්තමේන්තුව
3 2014.12.30
4 අභ්‍යන්තර අනුයුක්ත කිරීම .
5 ඔබගේ අංක DOL / ADM / 11/02/05 වන 2014.12.16 වන දින ලිපිය හා බැඳේ .
6 සංවර්ධන නිලධාරී එස් . ඩබ් . පී . එන් . විජේරත්න මහතා භාෂා විද්‍යාගාරයේ රාජකාරී භාර ගත් බවට මෙයින් දන්වා සිටිමි .
7 ඒ . එල් . නලන්ත .
8 භාෂා විද්‍යාගාර අධිකාරී

FIGURE 3.1: A sample screenshot where there are mismatches in the punctuation in parallel sentences.

TABLE 3.1: Statistics on parallel data (S- Sinhala, T- Tamil)

	Source type	no of sentences	no. of words (S)	no. of words (T)
Source-A	Official letters	9,151	103,864	94,513
Source-B	Circulars	1,588	27,592	22,568
Source-C	Annual Reports	2,088	27,074	23,054
Source-D	Order papers	9,194	160,776	133,464
Source-E	Estab. code	3,076	71,124	58,058

Apart from the target side of parallel data, freely available single source data was gathered from different sources, with the motive to reduce the sparseness in the language model. The sources of these data were wiki dumps, news articles, blog post, and bible translations. Table 3.2 and 3.3 illustrate the statistics on Sinhala, and Tamil monolingual data, respectively.

TABLE 3.2: Statistics on Sinhala monolingual data

	Source type	no. of sentences	no. of words
Source-X	Wiki dump [99]	200,000	3,247,421
Source-Y	News crawl	4,535,660	64,888,644

Table 3.3: Statistics on Tamil monolingual data

	Source type	no. of sentences	no. of words
Source-P	Bible, online blogs [100]	169,871	3,337,591
Source-Q	BBC news [101]	6,097	91,712
Source-R	Wiki dump [99]	300,000	7,649,915
Source-S	News crawl [99]	1,000,000	24,081,039
Source-T	IAS-NLP [102]	50,000	624,002
Source-U	Annual report	92,902	585,718

3.4 Selection of an SMT Framework

In order to select an SMT translation framework, freely available SMT systems were analyzed. The following systems were considered:

- Moses [103]
- Phrasal [104]
- Pharaoh [105]

Moses is the descendant of Pharaoh with more enhancements. Both were the research outcomes of the University of Edinburgh. Phrasal was the latest and it is from Stanford University. The initial attempt was carried out with Phrasal. However, the code base had many code breaks and the community support was very poor. In contrast, Moses community was very active and the code was maintained well with nightly builds. Therefore, Moses was selected as the SMT framework to develop the system.

3.5 Baseline System Setup

The SMT system was built using Moses. Figure 3.2 illustrates the baseline Si-Ta system.

Careful consideration was given when partitioning data for training, tuning, and testing. The standard Moses filtration was utilized to confirm that the sentences with extreme

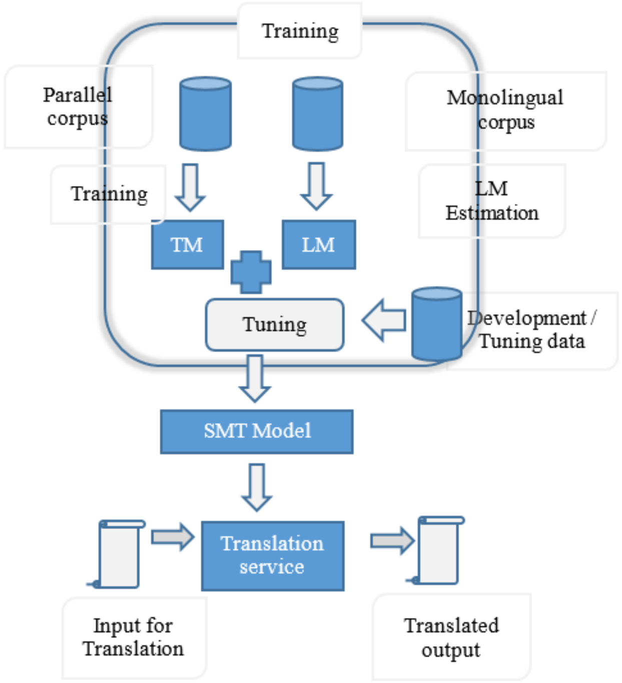


FIGURE 3.2: Baseline Si-Ta system

length ratio differences are removed effectively as this makes the word alignment task more complex. As the official letters had a high number of repeated phrases (same headers and footers were repeated over multiple documents), to make sure that testing and tuning will not be biased, a text similarity-based technique was used to extract unique sentences for testing and tuning. For testing and tuning, respectively 300, 1000 sentences were used while the test was used for training.

For word alignment, Giza++ [30] was used with default settings which are '*msd-bidirectional-fe*' as the reordering technique and '*grow-diag-final-and*' as the symmetrization heuristics.

As the smoothing technique, 'Good Turing' [2] was used to smooth the phrase table scores (smoothing techniques are a way to add relative frequency estimates to compensate data sparsity [106]). Lexical translation scores, phrase translation score, and a linear distortion score were used along with word and phrase penalties in the translation model as features. These features have shown favourable results for other language pairs in previous experiments.

SRILM [107] was used for language modeling. The target side of the parallel data and the target side monolingual data were used to build the language model. 3-gram back-off language models were created with modified Kneser-Ney [37] as smoothing technique. They were the commonly used features in many prior experiments.

For decoding, cube pruning technique with the maximum phrase length of 5 and a stack size of 5000 [108] was used in Moses. Minimum Error Rate Training (MERT) [46] was used for feature weight tuning using 100 best translations of the tuning set.

3.6 Summary

This chapter discussed the background details of building the baseline system along with the reasoning for selecting SMT as the implementing methodology. The chapter also described the data sources and technical details of implementation.

Chapter 4

Techniques for Improving SMT

4.1 Introduction

As explained in the previous chapter, the system was built using minimal resources. This creates abundant avenues for improvements. With the consideration on the limitations as well as the availability of resources, possible improvement avenues were evaluated on the baseline system.

Though the amount of available parallel data was considerably less, larger open domain monolingual data, bilingual terminology/glossaries, general dictionary entries, and larger name list in both the languages were attainable. By utilizing these resources, different techniques were evaluated to improve the translation quality by domain adaptation, integration of terminologies, dictionaries and name list, and handling the out-of-vocabulary using paraphrasing techniques, based on word embedding. Here a novel data augmentation technique is presented, which is used to generate synthetic parallel data out of available parallel data and a bilingual list. In addition, a heuristic based novel paraphrase technique is discussed under out-of-vocabulary (OOV) handling. This chapter elaborates on these techniques.

4.2 Domain Adaptation in SMT

As mentioned in the previous chapter, the system was built with a single translation model and a single language model. However, the writing style and the topic of interest differs from one data source to another. For example, the writing style of official letters is formal while the writing style in blog articles is more slang and colloquial while order papers were more like debate where sentences are questions and answers.

In the baseline system, no different priority was set for the data sources, according to their relevance to the domain of consideration. All data sources were used as a collection, giving the same emphasis over the translation.

With the intention of fine-tuning the system towards the domain, two different domain adaptation techniques were evaluated. They are:

- **Multiple models [2]:** Partitioning the data according to the relevance to the domain, and building individual models for each set.
- **Data filtration [70]:** Filter off the less relevant content from the corpus, based on similarity measures.

The following two sub-sections describe these two techniques.

4.2.1 Multiple Models

As mentioned in section 3.3, data was gathered from multiple different sources. Especially the monolingual data was from diverse domains. The gathered data was categorized into three categories, namely, *in-domain, pseudo-in-domain,* and *out-domain,* based on the context, relevance, and writing style.

Data gathered from official letters (e.g., Department of Official Languages, Secretariat department, etc.) was considered as *0*in-domain. The data from the other sources such as parliament order papers, annual reports, establishment codes, and circulars were categorized under *pseudo-in-domain.* Although these were the documents from government institutions, the style of writing was rather peculiar from the official letters which were considered as *in-domain* (e.g. The parliament order papers were more like debate while establishment codes had definitions and procedures).

Data collected from the web, (such as news articles, wiki dumps, and blogs), and data collected from free online sources were classified as *out-domain* data, as the context and writing style were quite different when compared to *in-domain* official government letters.

For language modeling, individual models were built for each category and two different integration techniques were evaluated:

- **Log-linear interpolation [109]:** The created individual models were added as sub-models to the system. At the initial stage, all these models were given the same weights. At the time of tuning, these weights were adjusted with regard to their applicability to the tuning set. Figure 4.1 illustrates the conceptual design of log-linear interpolation.
- **Linear interpolation [37]:** A new model is produced by combining a number of existing models in a specified ratio. The ratio is determined based on the perplexity measures for the tuning set. Here in the SMT system, only a single language model is used, which is an interpolated model. Figure 4.2 depicts the conceptual design of linear interpolation.

4.2.2 Data Filtering

Apart from segmenting the data and creating individual language models, as the ratio of *out-domain* data to *in-domain* data was considerably higher, data filtration techniques were evaluated over *out-domain* data.

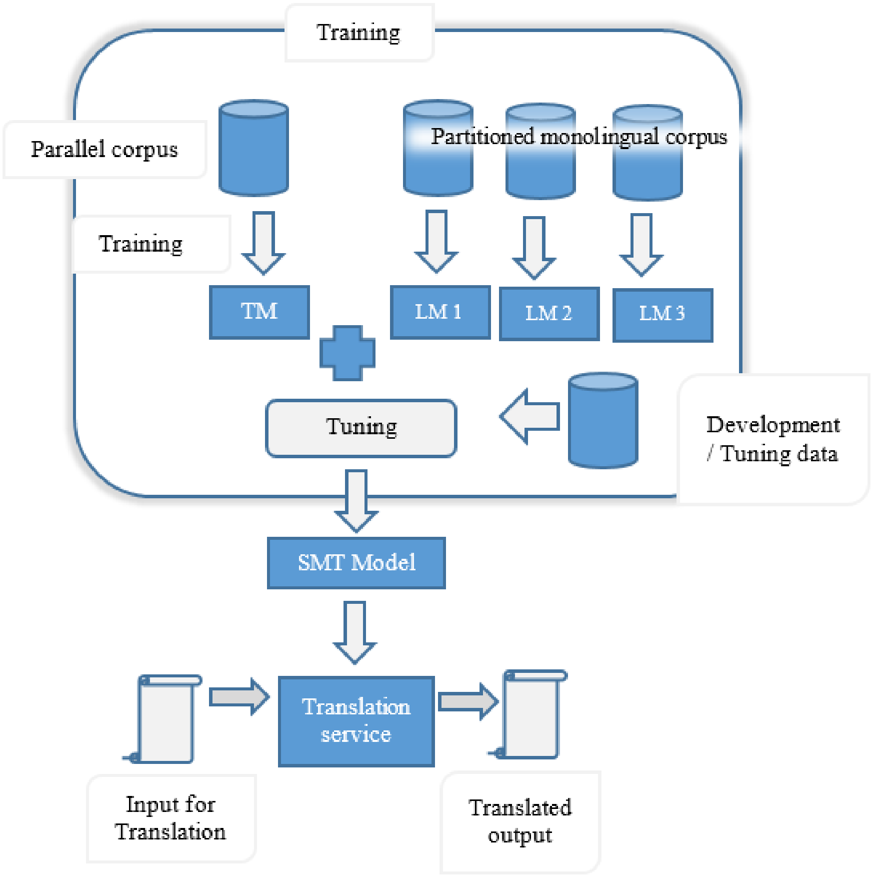


FIGURE 4.1: Language models are log-linearly interpolated. LM 1, LM 2 and LM 3 are the individual language models created from the *in-domain, pseudo-in-domain* and *out-domain* data.

XenC [110], a well-known open source data filtration tool was used to do the filtration over the *out-domain* data. Two separate LMs were created for *in-domain* and *out-domain* data using SRILM. These LMs were used in calculating the perplexity difference for each sentence, (in the *out-domain* corpus) between both the models. This difference between the perplexities is used in determining the eligibility of a sentence to be filtered out. The sentences with the value (difference in perplexity) higher than the threshold are filtered out.

The perplexity of each data type (domain based) and the filtered data are tabulated in Table 4.1 and 4.2 for Sinhala and Tamil data sources accordingly.

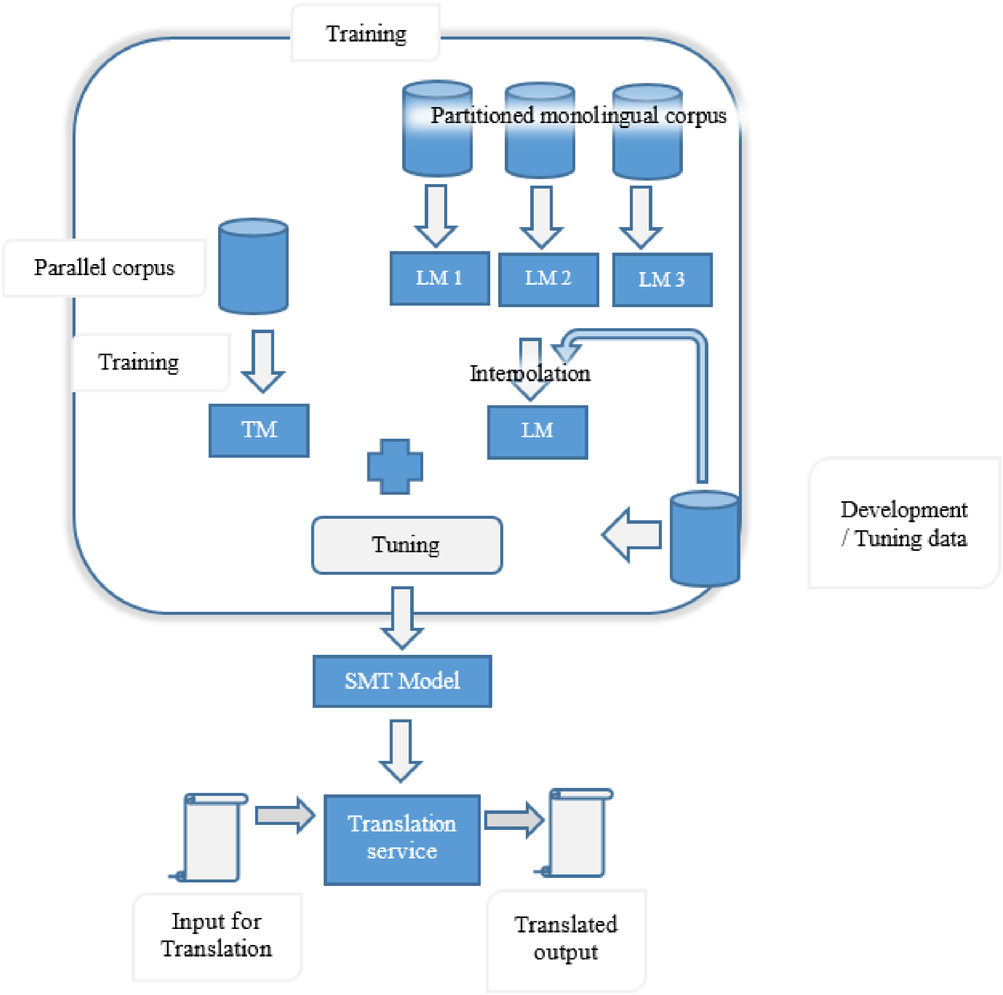


FIGURE 4.2: Language models are linearly interpolated. LM 1, LM 2 and LM 3 are the Language models created out of the *in-domain, pseudo-in-domain* and *out-domain* data while the LM is the interpolated language model.

This filtered *out-domain* corpus was also integrated into the system in linear and log-linear fashion.

4.3 Terminology Integration in SMT

With the intention to enhance the translation quality, two domain-specific bilingual lists were utilized. Both the collections were bilingual lists of terms in their base forms, which were created by professional translators. They are:

Table 4.1: Perplexity values for different domain data - Sinhala

Source	Perplexity
In-domain	87.42
Pseudo-in-domain	604.48
Out-domain	918.38
Filtered out-domain	518.18

Table 4.2: Perplexity values for different domain data - Tamil

Source	Perplexity
In-domain	214.62
Pseudo-in-domain	415.56
Out-domain	2210.49
Filtered out-domain	1814.93

- *TERM-1*: A list that consists of official designations and names of government organizations. The entries were 2-5 words length nominal phrases, where the average length was 3 words long.
- *TERM-2*: A list of glossary related to government administration and operations. The terms come from public administration, land administration, financial regulations, Army, Navy, Air Force and Police. The entries were phrases with a length of 1-3 words which include verbs, nouns, adjectives, and adverbs.

Entry counts on each lists is tabulated in Table 4.3. These lists were integrated using four different static and a dynamic integration techniques. In static integration, three techniques have been attempted in prior researches, while one is a novel technique where the bilingual list is used for generating augmented parallel data.

For each of the integration techniques explained here in this sub-section, these lists were incorporated one by one individually. Therefore, the term *Terminology* in the rest of this section refers to the list *TERM-1* and *TERM-2* in general. Experiments for each technique were repeated for each of the lists.

The following subsections elaborate on each of these techniques.

Table 4.3: Statistics on the terminology list utilized

Source	No of entries
TERM-1	5,291
TERM-2	19,861

4.3.1 Static Integration Techniques

Under this sub-category, four different terminology integration techniques were evaluated. As these techniques impact the pre-defined models and require re-training, these techniques are referred to as *static integration techniques* [72].

4.3.1.1 As Corpus

This is the simplest form of the static integration technique [75]. In this technique, to train the translation model, *Terminology* is used along with the parallel data. The entries in the target side of the *Terminology* are used along with the monolingual data in training the language model. Concept of this technique can be depicted as in Figure 4.3.

This technique is quite effective because this technique increases the term coverage of the models (reduce the chance of a term being out-of-vocabulary). This creates the likelihood of having a minimum of one translation hypothesis for the term. This technique works better when the term is found in both, the parallel corpus and the monolingual corpus. At the time of translation, the translation model is accountable for producing translation hypotheses. Then the language model is accountable for selecting the best option from the options produced by the translation model, by estimating their presence in the target language. Especially for terms with multiple words, this plays a vital role in selecting the most suitable word combination. For example: for the word 'පොලිස් ස්ථානය' /police stanaya/ (police station), the correct Tamil translation is '**பொலிஸ் நிலையம்**'/police nilayam/. However the word 'ස්ථානය' /sthanaya/ alone refers to 'place' for which the corresponding Tamil word is '**இடம்**' /idam/. When the term '**பொலிஸ் நிலையம்**' /police nilayam/ is not present in the language model, though the

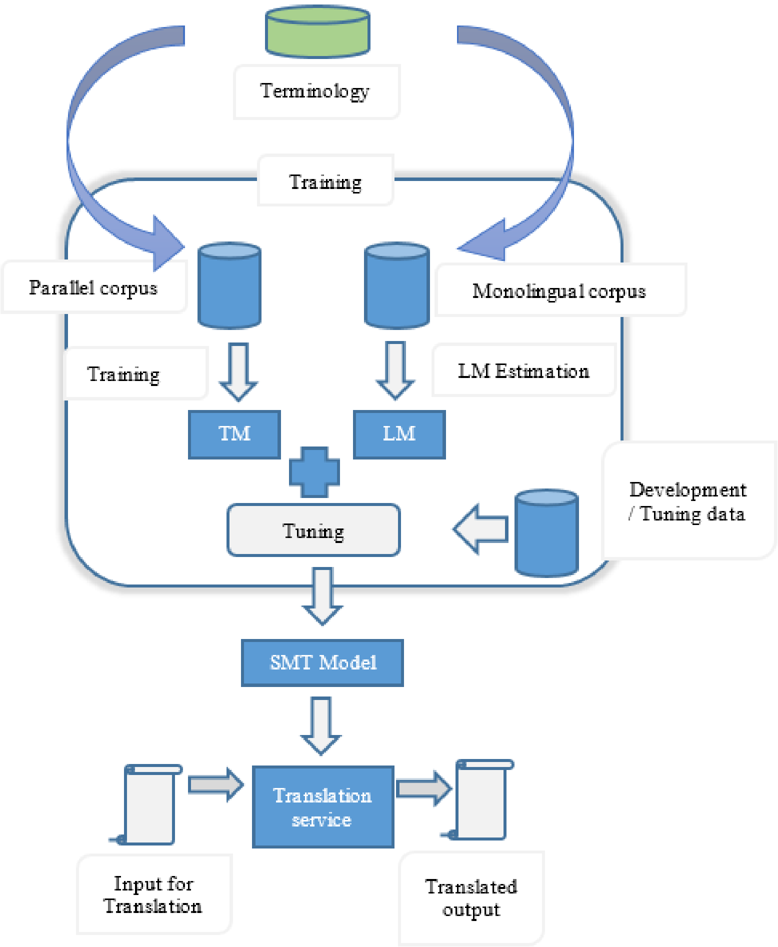
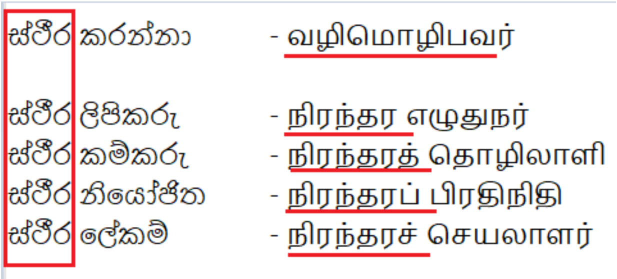


FIGURE 4.3: Conceptual design of terminology integration-‘as corpus’

correct translation ‘**பொலிஸ் நிலையம்**’ for ‘පොලිස් ස්ථානය’ had the highest translation probability as the whole phrase, the system does not pick it. Instead, it provides the translation as ‘**பொலிஸ் இடம்**’ /police idam/ which is incorrect.

4.3.1.2 Multiple Tables

In this technique, similar to an approach evaluated by Ren et al. [73], two different phrase tables (translation models) were created. One is for the parallel corpus and the other is for *Terminology*. However, in Ren et al. [73]'s approach, in the second phrase table created from *Terminology*, the probabilities are set to 1. In our approach, both the phrase tables were generated using the default training procedure, where heuristics were used for word alignment. The reason is that the *Terminology* consists of many multi-word terms. Therefore, setting the probabilities to '1' may cause problems when the same word has different translations in different term entries according to the context. Figure 4.4 illustrates one such example. In general, the word 'ස්ථීර' /istheera/ (permanent) is referred to as 'நிரந்தரம்' /nirandaram/. However, the first example illustrated in Figure 4.4 refers to 'one who second it', and the rest of the examples have the inflection variation of the word 'permanent'.



ස්ථීර කරන්නා	- வழிமொழிபவர்
ස්ථීර ලිපිකරු	- நிரந்தர எழுதுநர்
ස්ථීර කම්කරු	- நிரந்தரத் தொழிலாளி
ස්ථීර නියෝජිත	- நிரந்தரப் பிரதிநிதி
ස්ථීර ලේකම්	- நிரந்தரச் செயலாளர்

FIGURE 4.4: Sinhala word 'ස්ථීර' having translation variations based on the context and inflection as highlighted.

Both the phrase tables (the table created from the parallel corpus and the table built using *Terminology*) were used as two sub-models in the log-linear model. The initial weights for both the tables were set to the same value. At the time of tuning, the weights of both phrase tables were adjusted with reference to the tuning set. The abstract design of this integration technique is illustrated in figure 4.5. Here TM-1 refers to the primary phrase table created from the parallel data, while TM-2 refers to the phrase table created from *Terminology*.

4.3.1.3 Merged Tables

This technique follows the same approach as 'fill-up' by Arcan et al. [78]. Individual phrase tables were created for parallel data and *Terminology* separately, as in section 4.3.1.2. The phrase table from the parallel data is considered as the primary. From

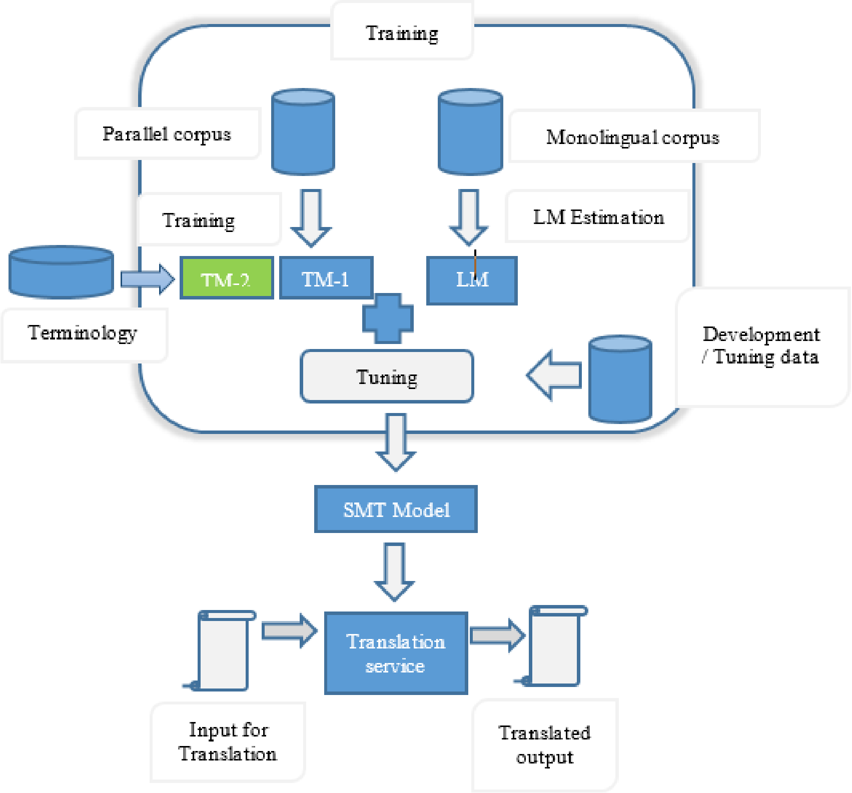
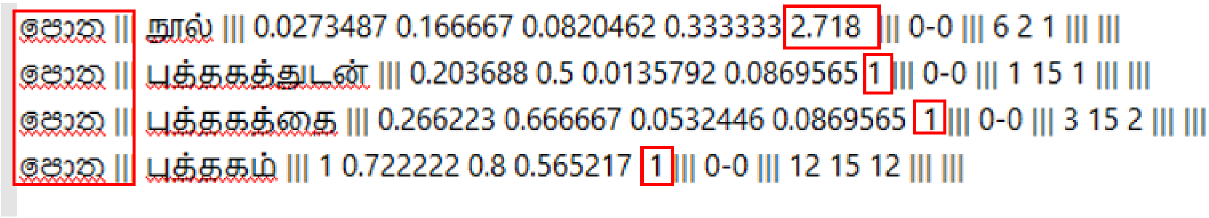


FIGURE 4.5: Conceptual design of terminology integration technique - 'Multiple tables'

the secondary phrase table (phrase table from *Terminology,*) only the items missing in the primary table are added to the newly created table. Apart from the default features in the phrase table, a new feature was included to the table to differentiate the fill-up entries from the original entries. At the time of tuning, this feature weight was adjusted along with the other features. Therefore, based on the impact of the added entries (secondary table entries included in the new phrase table) on the tuning set, the weight is either increased or reduced. This technique basically uses the table merge feature in Moses [111] to merge the individual tables into one final table. Figure 4.6 is a sample screen-shot of the new phrase table with the new feature highlighted, where the binary values are set to their corresponding log-values (1 and 2.718).



```
පොත || நூல் ||| 0.0273487 0.166667 0.0820462 0.333333 2.718 ||| 0-0 ||| 6 2 1 ||| |||
පොත || புத்தகத்துடன் ||| 0.203688 0.5 0.0135792 0.0869565 1 ||| 0-0 ||| 1 15 1 ||| |||
පොත || புத்தகத்தை ||| 0.266223 0.666667 0.0532446 0.0869565 1 ||| 0-0 ||| 3 15 2 ||| |||
පොත || புத்தகம் ||| 1 0.722222 0.8 0.565217 1 ||| 0-0 ||| 12 15 12 ||| |||
```

FIGURE 4.6: Sample screenshot of a phrase table for ‘merge-table’. Values for the new feature are highlighted.

4.3.1.4 Parallel Data Augmentation with Bilingual Lists

In the techniques mentioned in the previous sub-sections (section 4.3.1.1, 4.3.1.2 and 4.3.1.3), no contextual information for the terminologies is provided. In most of the previous work, this kind of lists were derived from the parallel corpus itself. However, for a low-resource setup, such as Sinhala-Tamil, the amount of parallel data is considerably lower. The lists used in this thesis were manually compiled by human translators. Therefore, with the motive to provide context and multiple occurrences for the terms, three techniques were proposed on data augmentation using the bilingual list and parallel data.

The following four subsections provide a detailed description of the augmentation techniques proposed by us. Experiments for these techniques were carried on only with ‘*TERM-1*’ as the type of words in this list was predictable (all the entries were nouns - either designations or organization names) while other list had entries of different word classes (nouns, verbs, adjectives, and adverb).

As the initial step of the first two techniques below, possible word embedding techniques were analyzed to identify the best suiting one. Two well-known techniques, namely Word2vec [112] and fast-text [113], were taken into consideration. Models were built for Sinhala, as well as for Tamil. An arbitrary set of words from Sinhala as well in Tamil were used to get their most similar words from the built vector spaces based on cosine similarity, for analysis. Figures 4.7, 4.8 are comparisons between two models for Sinhala and Tamil, respectively. As it is shown in these figures, the similar words based on fast-text are more towards the morpheme-based similarity (edit-distance), while Word2vec results are more towards the literal meaning of words (the reason may be that fast-text considers character level details in addition to the context when positioning the vectors). Due to this better result of Word2vec, a skip-gram word embedding model was created using Word2vec with the window size of 5 and vector dimension of 100, while considering a minimum of 3 occurrences of the word in the corpus for the word to be considered in the model. Here, values for the window size and dimension were considered based on common convention, while the occurrence was set to a low-value to have a higher vocabulary coverage.

1. **Based on ending word**

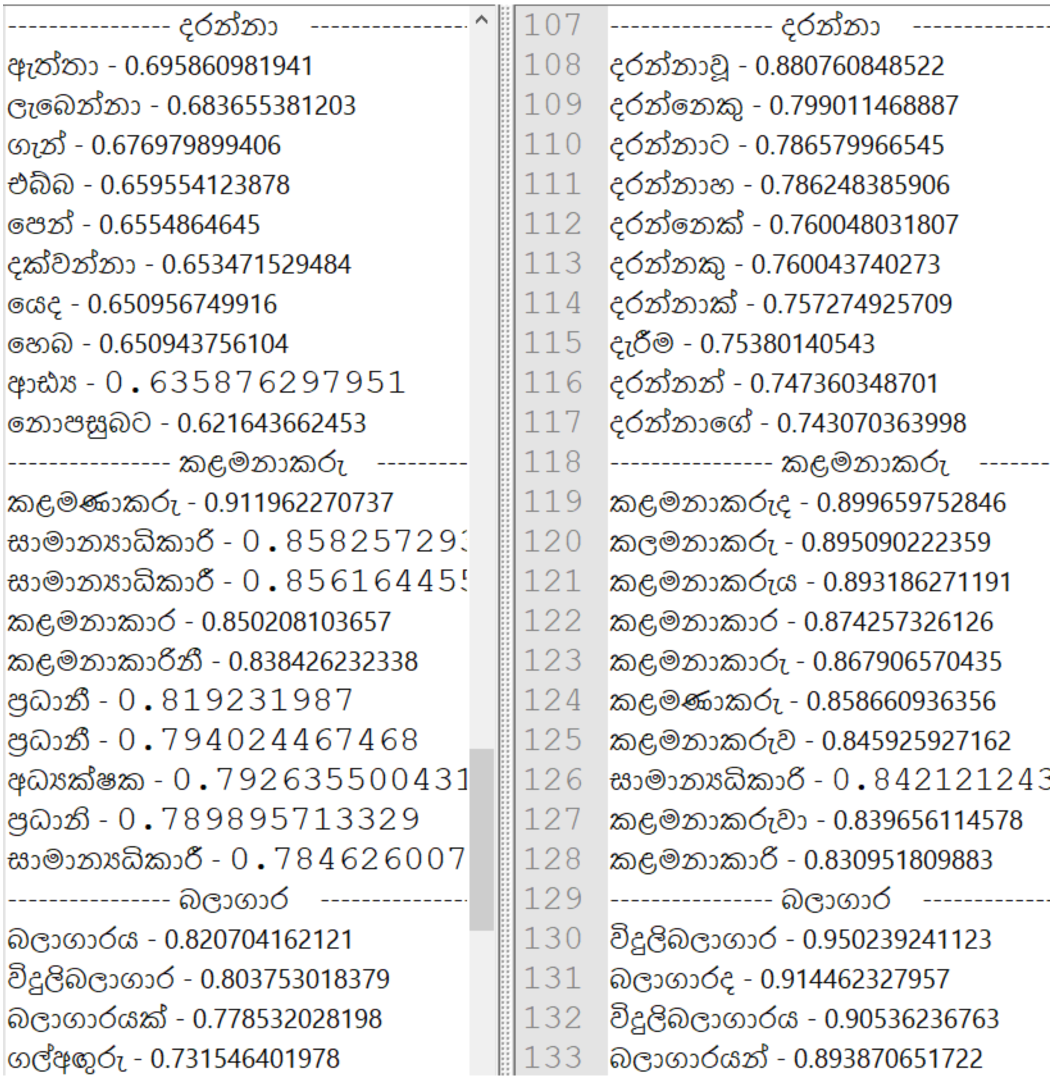
   The source side (Sinhala) of the bilingual list was clustered based on the word ending. For example, the words that have the ending as 'දෙපාර්තමේන්තුව' /departhamenthuwa/ (department) were clustered together. Therefore, this cluster consists of terms as of in Figure 4.9:

   For each word ending, a list of the most similar 10 words (similar list) along with the similarity index is being retrieved from the built word embedding model. At the same time, the same ending is added as the first entry to this list with the similarity index set to 1.

   For example, for the word 'දෙපාර්තමේන්තුව' the similarity is shown in Figure 4.10.

   Then for each headword entry in the ending word list, the occurrence of the same word or similar word in the parallel corpus is traced, based on the following technique. Algorithm 1 gives an abstract sketch of this proposed implementation.

   - Randomly a sentence is picked from the target side of the parallel data (in the above case, in the Sinhala side of parallel data) and check if the sentence consists of the occurrence of the same ending. (as in the above case, word 'දෙපාර්තමේන්තුව' is searched)



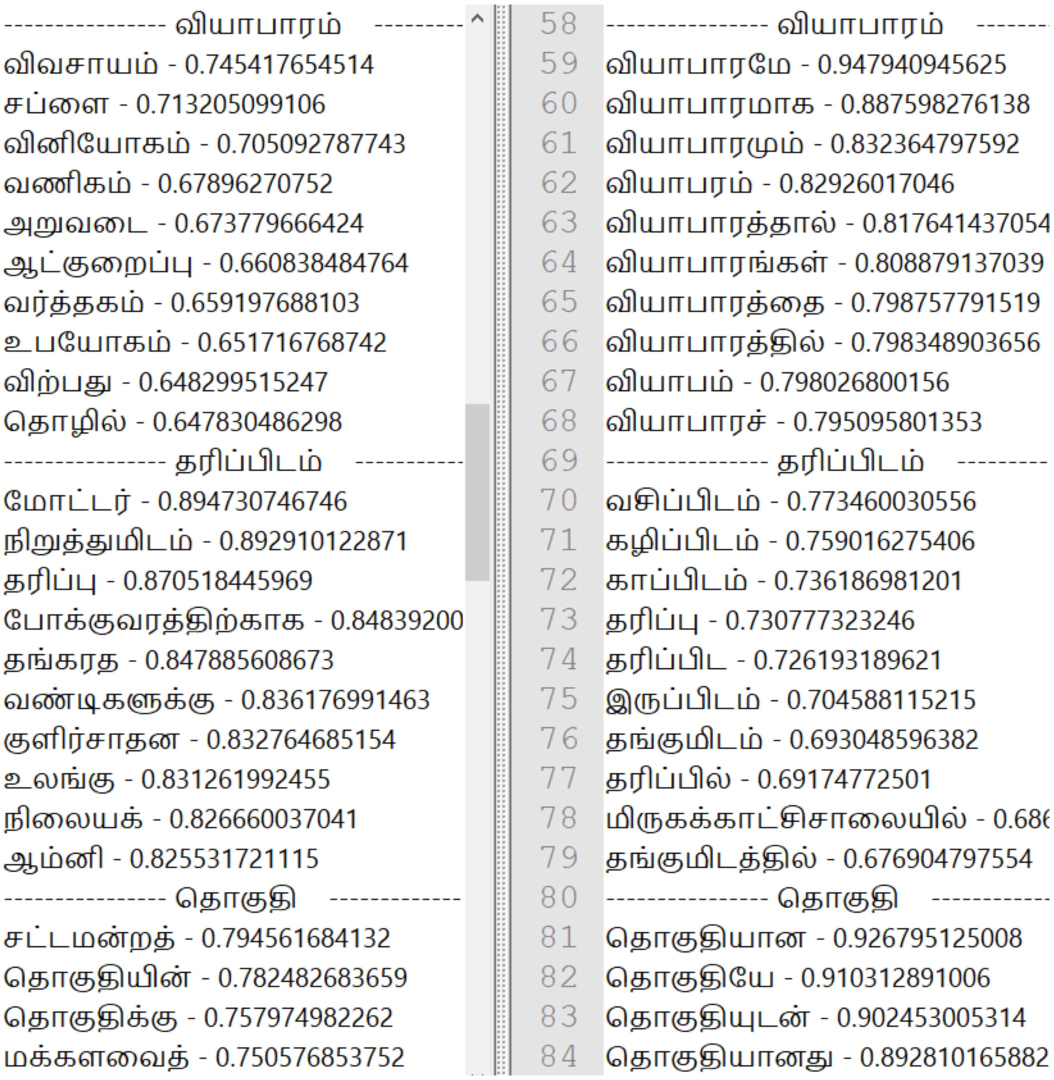
```
---------------- දරන්නා    ----------------
ඇත්තා - 0.695860981941
ලැබෙන්නා - 0.683655381203
ගැත් - 0.676979899406
එබ්බ - 0.659554123878
පෙන් - 0.6554864645
දක්වන්නා - 0.653471529484
යෙද - 0.650956749916
හෙබ - 0.650943756104
ආඪ්‍ය - 0.635876297951
නොපසුබට - 0.621643662453
---------------- කළමනාකරු    ---------
කළමණාකරු - 0.911962270737
සාමාන්‍යාධිකාරි - 0.85825729
සාමාන්‍යාධිකාරී - 0.85616445
කළමනාකාර - 0.850208103657
කළමනාකාරිනී - 0.838426232338
ප්‍රධානී - 0.819231987
ප්‍රධානී - 0.794024467468
අධ්‍යක්ෂක - 0.792635500431
ප්‍රධානි - 0.789895713329
සාමාන්‍යධිකාරී - 0.784626007
---------------- බලාගාර    --------------
බලාගාරය - 0.820704162121
විදුලිබලාගාර - 0.803753018379
බලාගාරයක් - 0.778532028198
ගල්අඟුරු - 0.731546401978
```

```
---------------- දරන්නා    --------------
දරන්නාවූ - 0.880760848522
දරන්නෙකු - 0.799011468887
දරන්නාට - 0.786579966545
දරන්නාහ - 0.786248385906
දරන්නෙක් - 0.760048031807
දරන්නකු - 0.760043740273
දරන්නාක් - 0.757274925709
දැරීම - 0.75380140543
දරන්නන් - 0.747360348701
දරන්නාගේ - 0.743070363998
---------------- කළමනාකරු    -------
කළමනාකරුද - 0.899659752846
කලමනාකරු - 0.895090222359
කළමනාකරුය - 0.893186271191
කළමනාකාර - 0.874257326126
කළමනාකාරු - 0.867906570435
කළමණාකරු - 0.858660936356
කළමනාකරුව - 0.845925927162
සාමාන්‍යධිකාරි - 0.84212124
කළමනාකරුවා - 0.839656114578
කළමනාකාරි - 0.830951809883
---------------- බලාගාර    ------------
විදුලිබලාගාර - 0.950239241123
බලාගාරද - 0.914462327957
විදුලිබලාගාරය - 0.90536236763
බලාගාරයන් - 0.893870651722
```

FIGURE 4.7: Comparison between similar word lists for sample Sinhala words based on Word2vec and fast-text based models. The left side list is fetched from the Word2vec model while the right side one is from the fastText.

- If found, the corresponding parallel pair is picked and the ending word is replaced with the term pair in the source sentence and then in target sentence. The word alignment information (from Giza++ output) is used to find the corresponding target side word to be replaced.

  Figure 4.11 illustrates one such example for the term pair 'රාජ්‍ය ණය දෙපාර්තමේන්තුව' /raajhya naya departhamenthuwa/, '**அரசாங்கக் கடன் திணைக்களம்**' /arasaangak kadan thinaikkalam/ (Public Debt Department). Here the first pair is the original and the second is the augmented pair.



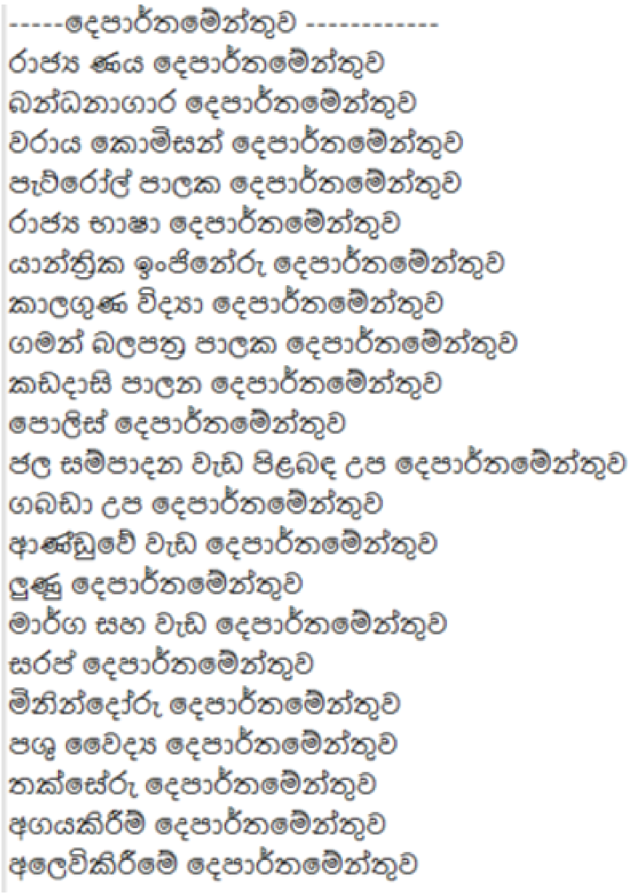
```
---------------- வியாபாரம்    ---------
விவசாயம் - 0.745417654514
சப்ளை - 0.713205099106
வினியோகம் - 0.705092787743
வணிகம் - 0.67896270752
அறுவடை - 0.673779666424
ஆட்குறைப்பு - 0.660838484764
வர்த்தகம் - 0.659197688103
உபயோகம் - 0.651716768742
விற்பது - 0.648299515247
தொழில் - 0.647830486298
---------------- தரிப்பிடம்    ----------
மோட்டர் - 0.894730746746
நிறுத்துமிடம் - 0.892910122871
தரிப்பு - 0.870518445969
போக்குவரத்திற்காக - 0.84839200
தங்கரத - 0.847885608673
வண்டிகளுக்கு - 0.836176991463
குளிர்சாதன - 0.832764685154
உலங்கு - 0.831261992455
நிலையக் - 0.826660037041
ஆம்னி - 0.825531721115
---------------- தொகுதி    -------------
சட்டமன்றத் - 0.794561684132
தொகுதியின் - 0.782482683659
தொகுதிக்கு - 0.757974982262
மக்களவைத் - 0.750576853752
```

```
---------------- வியாபாரம்    -------
வியாபாரமே - 0.947940945625
வியாபாரமாக - 0.887598276138
வியாபாரமும் - 0.832364797592
வியாபரம் - 0.82926017046
வியாபாரத்தால் - 0.817641437054
வியாபாரங்கள் - 0.808879137039
வியாபாரத்தை - 0.798757791519
வியாபாரத்தில் - 0.798348903656
வியாபம் - 0.798026800156
வியாபாரச் - 0.795095801353
---------------- தரிப்பிடம்    ---------
வசிப்பிடம் - 0.773460030556
கழிப்பிடம் - 0.759016275406
காப்பிடம் - 0.736186981201
தரிப்பு - 0.730777323246
தரிப்பிட - 0.726193189621
இருப்பிடம் - 0.704588115215
தங்குமிடம் - 0.693048596382
தரிப்பில் - 0.69174772501
மிருகக்காட்சிசாலையில் - 0.68
தங்குமிடத்தில் - 0.676904797554
---------------- தொகுதி    ------------
தொகுதியான - 0.926795125008
தொகுதியே - 0.910312891006
தொகுதியுடன் - 0.902453005314
தொகுதியானது - 0.892810165882
```

FIGURE 4.8: Comparison between similar word lists for sample Tamil words based on Word2vec and fast-text based models. The left side list is fetched from the Word2vec model while the right side one is from the fastText.

This new sentence pair generation takes place for each term in that same cluster. However, to make sure that the same sentence pair is not utilized with multiple entries with the same ending, for each term, the next occurrence in the loop is utilized.

- Else, until the occurrence is found in the parallel data, iterate through the parallel data.
- If there is no occurrence found, the same search operation (mentioned above) is done with the next word in the similarity list and is continued until either

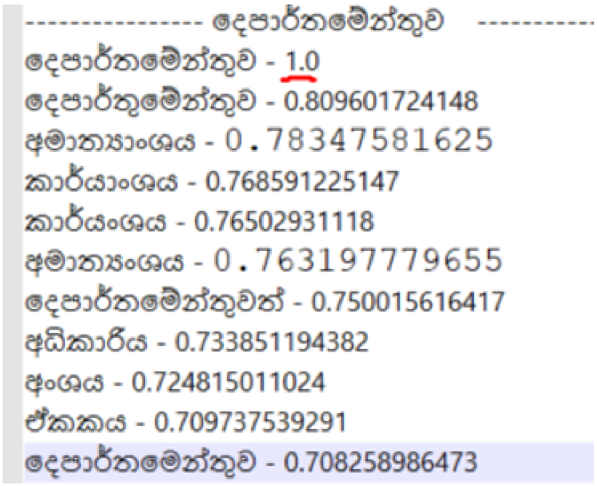


-----දෙපාර්තමේන්තුව ------------
රාජ්‍ය ණය දෙපාර්තමේන්තුව
බන්ධනාගාර දෙපාර්තමේන්තුව
වරාය කොමිසන් දෙපාර්තමේන්තුව
පැට්රෝල් පාලක දෙපාර්තමේන්තුව
රාජ්‍ය භාෂා දෙපාර්තමේන්තුව
යාන්ත්‍රික ඉංජිනේරු දෙපාර්තමේන්තුව
කාලගුණ විද්‍යා දෙපාර්තමේන්තුව
ගමන් බලපත්‍ර පාලක දෙපාර්තමේන්තුව
කඩදාසි පාලන දෙපාර්තමේන්තුව
පොලිස් දෙපාර්තමේන්තුව
ජල සම්පාදන වැඩ පිළබඳ උප දෙපාර්තමේන්තුව
ගබඩා උප දෙපාර්තමේන්තුව
ආණ්ඩුවේ වැඩ දෙපාර්තමේන්තුව
ලුණු දෙපාර්තමේන්තුව
මාර්ග සහ වැඩ දෙපාර්තමේන්තුව
සරජ් දෙපාර්තමේන්තුව
මිනින්දෝරු දෙපාර්තමේන්තුව
පශු වෛද්‍ය දෙපාර්තමේන්තුව
තක්සේරු දෙපාර්තමේන්තුව
අගයකිරීම් දෙපාර්තමේන්තුව
අලෙවිකිරීමේ දෙපාර්තමේන්තුව

FIGURE 4.9: A sample cluster of terms based on the ending word

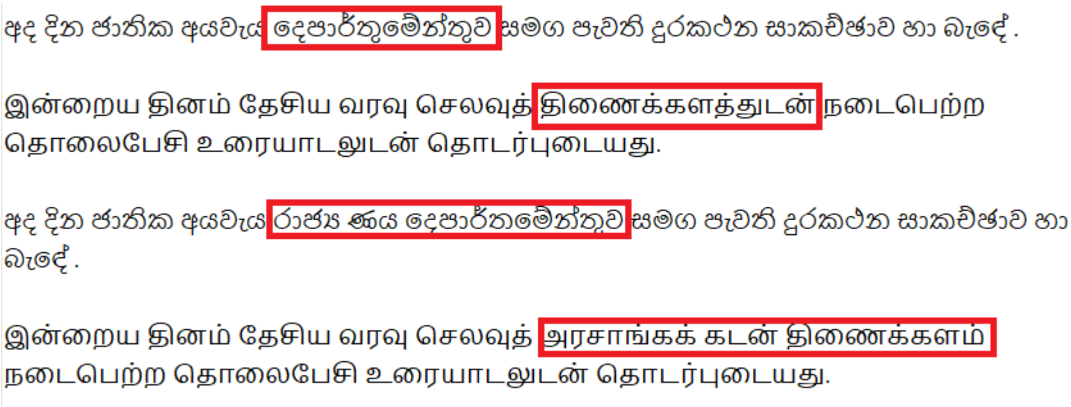
it finds an occurrence or reach the end of the search, for each word in the similar list.

- If an occurrence is found, parallel data is generated.
- Else the terms in the cluster are not used in data generation.



```
---------------- දෙපාර්තමේන්තුව   -----------
දෙපාර්තමේන්තුව - 1.0
දෙපාර්තුමේන්තුව - 0.809601724148
අමාත්‍යාංශය - 0.78347581625
කාර්යාංශය - 0.768591225147
කාර්යංශය - 0.76502931118
අමාත්‍යංශය - 0.763197779655
දෙපාර්තමේන්තුවත් - 0.750015616417
අධිකාරීය - 0.733851194382
අංශය - 0.724815011024
ඒකකය - 0.709737539291
දෙපාර්තමේන්තුව - 0.708258986473
```

FIGURE 4.10: A sample similarity list retrieved based on word embedding



අද දින ජාතික අයවැය දෙපාර්තුමේන්තුව සමග පැවති දුරකථන සාකච්ඡාව හා බැදේ .

இன்றைய தினம் தேசிய வரவு செலவுத் திணைக்களத்துடன் நடைபெற்ற தொலைபேசி உரையாடலுடன் தொடர்புடையது.

අද දින ජාතික අයවැය රාජ්‍ය ණය දෙපාර්තමේන්තුව සමග පැවති දුරකථන සාකච්ඡාව හා බැදේ .

இன்றைய தினம் தேசிய வரவு செலவுத் அரசாங்கக் கடன் திணைக்களம் நடைபெற்ற தொலைபேசி உரையாடலுடன் தொடர்புடையது.

FIGURE 4.11: An original (first pair) and the augmented sentence pair (second pair) based on *'Based on ending word'* technique. The term replaced is highlighted.

Algorithm 1: Data augmentation for *'Based on ending word'*

```
Data: parallelData, wordAlignment, bilingualList, embeddingModel
Result: augmented parallel data
$Cl = ClusterBilingualListOnEndingWord(bilingualList)$;
$SiMap = SimilarListForEndingWord(bilingualList, embeddingModel, Cl)$;
for each $key_i$, $list_i$ in Cl do
    for each $term$ in $list_i$ do
        $siList = SiMap_{key_i}$;
        $candidateSentence =$
          $randomOccuranceOfSimilarWordInCorpus(parallelData, siList)$;
        $augmentedSentence =$
          $generateSentence(candidateSentence, wordAlignment, term)$ ;
    end
end
```

In this technique, the ending word or the word similar to the ending word is replaced by the term (in most of the case a multi-word expression).

2. **Based on ending word + POS**

   This technique is an advanced form of the previous *'Based on ending word'* technique. Instead of replacing only the end word, a relevant multi-word term is identified and replaced. In order to detect the term boundary, along with the *'Based on ending word'* technique, POS-tagged data is deployed.

   Before processing, parallel data is tagged for POS using a tagger [114]. The POS patterns for the terms are manually observed and studied. Based on the analysis, the following regular expression was identified as the POS pattern of multi-word terms that were found in the parallel corpus (the first sentence in Figure 4.12 highlights one such example):

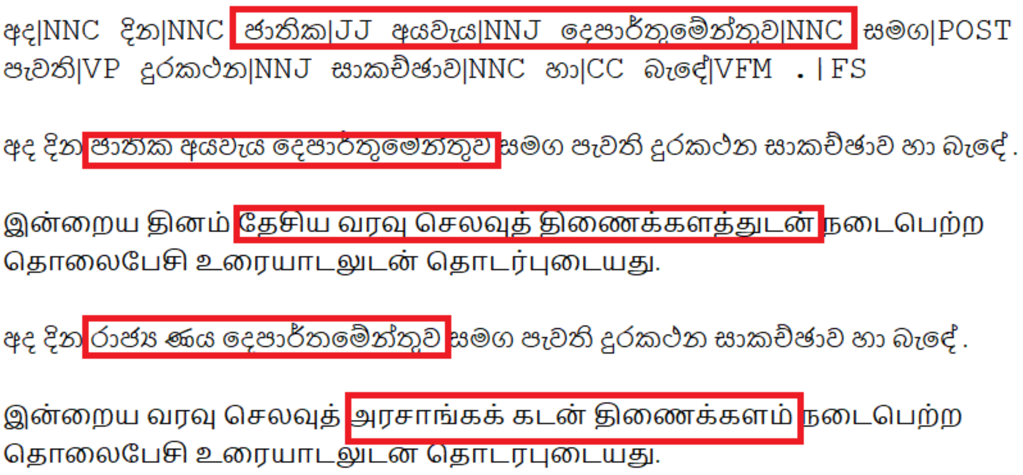
   ('JJ')* ['NNP' | 'NNC' | 'NNJ']+

JJ - adjective

NNP - Proper noun

NNC - Common noun

NNJ - Adjective noun



අද|NNC දින|NNC ජාතික|JJ අයවැය|NNJ දෙපාර්තුමේන්තුව|NNC සමග|POST පැවති|VP දුරකථන|NNJ සාකච්ඡාව|NNC හා|CC බැදේ|VFM .|FS

අද දින ජාතික අයවැය දෙපාර්තුමේන්තුව සමග පැවති දුරකථන සාකච්ඡාව හා බැදේ.

இன்றைய தினம் தேசிய வரவு செலவுத் திணைக்களத்துடன் நடைபெற்ற தொலைபேசி உரையாடலுடன் தொடர்புடையது.

අද දින රාජ්‍ය ණය දෙපාර්තමේන්තුව සමග පැවති දුරකථන සාකච්ඡාව හා බැදේ.

இன்றைய வரவு செலவுத் அரசாங்கக் கடன் திணைக்களம் நடைபெற்ற தொலைபேசி உரையாடலுடன தொடரபுடையது.

Figure 4.12: Boundary detection based on *'Based on ending word + POS'* for the same example illustrated in 4.11.

Based on this POS pattern, data generation technique works as follows:

- Cluster the term list and search for the occurrence, as in the section *'Based on ending word'*.
- When a candidate sentence is found, in addition to the above search criteria, the POS-tag of the word is consulted. If the POS is either 'NNP' or 'NNC' or 'NNJ', the sentence is considered, otherwise rejected.
- Then the candidate sentence is further analyzed to identify the boundary of the term. This is done purely based on the POS-tags of the words that occur before the candidate word in the sentence. The POS pattern as of the above mentioned regular expression is searched through the sentence in reverse order. If this pattern is broken, that point is identified to be the beginning of the term boundary.

- Based on the word alignment information, the corresponding translation of the candidate term is identified from the other side of the parallel corpus.
- Then the terms in both sides are replaced by the bilingual pair.

For the same example illustrated for the *'Based on ending word'* technique, Figure 4.12 shows the augmented sentence for '*Based on ending word + POS*' along with the term boundary based on POS highlighted.

3. **Based on ending word + improved POS**

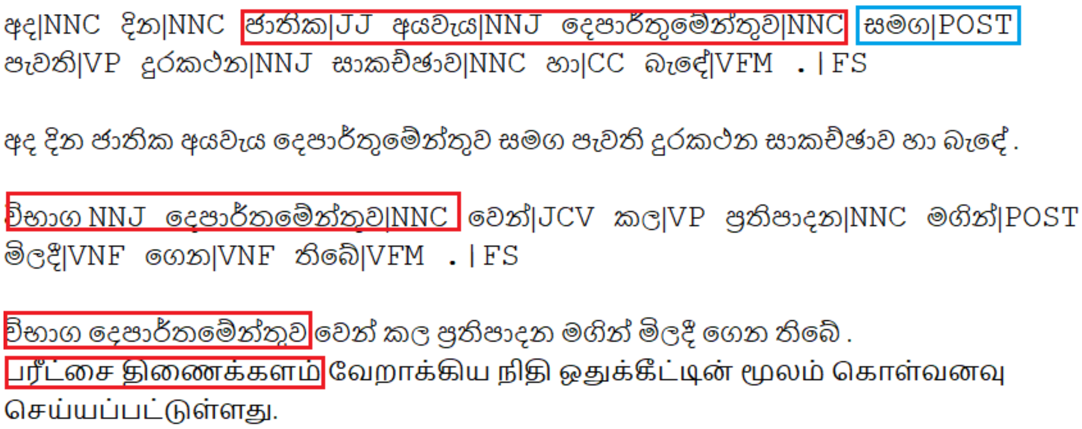
   This technique follows the same procedure as the technique *'Based on ending word + POS'*. However, an extra step of verification is done at the point of selecting the candidate sentence pair. The POS-tag of the word that follows the candidate term is consulted. If the tag is 'POST', that candidate is omitted.

   This heuristic was imposed based on manual observation. For each word with the tag 'POST', the preposition is agglomerated to a word in Tamil. (words that fall under the tag 'POST ' are: මගින්, සිට, සඳහා, අනුව, විසින්, බැවින්, ලෙස, මෙන්, වෙත, හා, which are the prepositions to which the Tamil equivalents are agglomerated with the base word). Figure 4.13 illustrates two sample Sinhala sentences. Here the same example that is used in the previous two techniques is disqualified due to the new rule over the POS-tag of the word followed.

4. **Based on NER**

   As the list utilized (TERM-1) in data augmentation consisted only of names of organizations and designations, with the motive to find better data augmentation, NE tagged (only source side - Sinhala) parallel data is utilized. The procedure is as follows.

   - Since the list consisted of two types of NEs (designations and organizations), the list was manually split into two lists, as designations and organizations.
   - Sinhala side (source) of parallel data was tagged with NEs using a Named Entity Recognizer (NER) [114]



අද|NNC දින|NNC ජාතික|JJ අයවැය|NNJ දෙපාර්තුමේන්තුව|NNC සමග|POST පැවති|VP දුරකථන|NNJ සාකච්ඡාව|NNC හා|CC බැදේ|VFM .|FS

අද දින ජාතික අයවැය දෙපාර්තුමේන්තුව සමග පැවති දුරකථන සාකච්ඡාව හා බැදේ .

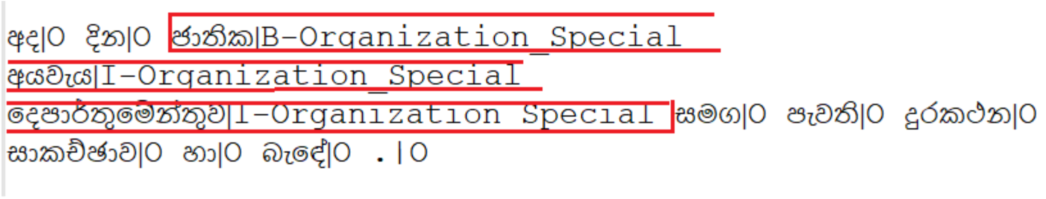
විභාග NNJ දෙපාර්තමේන්තුව|NNC වෙන්|JCV කල|VP ප්‍රතිපාදන|NNC මගින්|POST මිලදී|VNF ගෙන|VNF තිබේ|VFM .|FS

විභාග දෙපාර්තමේන්තුව වෙන් කල ප්‍රතිපාදන මගින් මිලදී ගෙන තිබේ .
பரீட்சை திணைக்களம் வேறாக்கிய நிதி ஒதுக்கீட்டின் மூலம் கொள்வனவு செய்யப்பட்டுள்ளது.

FIGURE 4.13: Example to illustrate the difference in *‘Based on ending word + improved POS’* and *‘Based on ending word + POS’*. The sentence what is being used in the *‘Based on ending word + POS’* technique does not meet the requirement of *‘Based on ending word + improved POS’* technique since the POS of the word that follows is ‘POST’.

- Data was tagged based on BIO (Beginning, Intermediate, and Out) standard. The tags used in this NER system were domain relevant (Organization_Generic (Government organizations), Organization_Special (Special names usually transliterate), Designation, Location, Person, Temporal, Other) as the tags are customized for official government documents.
- For each term in the list, randomly a sentence is picked from the source side (Sinhala) of parallel data. NE tags are checked for the relevant NE type (if the term is a designation, ‘Designation’ tag is looked for. If the term is an organization, ‘Organization_Generic’ is looked for).
- If the desired tag is found, then the corresponding target word (Tamil correspondence) is found using word alignment as in the previous three techniques. Then the identified word pair (in the candidate sentence) is replaced by the term pair (term and its corresponding translation from the list) and a new sentence is generated.
- Else, keep on iterating till a matching instance is found and generate a new parallel pair.

Figure 4.14 highlights how the term is detected based on the NE tag for the same example illustrated in Figures 4.11, and 4.12.



```
අද|O දින|O ජාතික|B-Organization_Special
අයවැය|I-Organization_Special
දෙපාර්තුමේන්තුව|I-Organization_Special සමග|O පැවති|O දුරකථන|O
සාකච්ඡාව|O හා|O බැදේ|O .|O
```

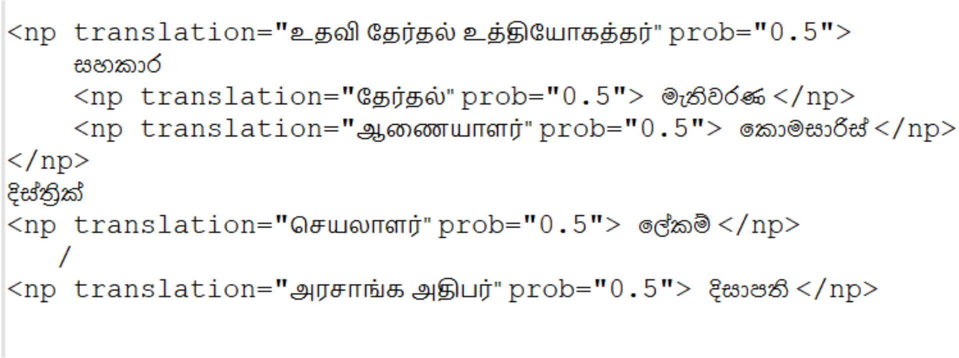
FIGURE 4.14: A sample sentence where the identification of the term is based on the NE tag. Term boundary is highlighted.

4.3.2 Dynamic Integration Technique

Integrating terminology to the SMT system at the time of training (static integration techniques) helps to adapt the system to the domain, however, with a cost of retraining the system partially or as a whole (either from re-training the models or re-tuning the weights). In dynamic integration, these changes are not required in the models, yet the text to be translated need to be pre-processed. Similar kind of dynamic integration technique Arcan et al. [78] was explored under this section. The following are the pre-processing steps in dynamic integration:

- The initial step is to find the terms in the source text. A simple list lookup against the *Terminology* is used to identify the terms in the test data.
- Once the term is identified, the term is enclosed within XML tags that adhere to the Moses XML format. Here the input text is externally enriched with the translation option based on *Terminology*. All these external translation options are provided with a translation probability of '0.5'. The figure 4.15 depicts a sample input text after the XML pre-processing. The words/phrases that were found in the *Terminology* are enclosed within *"<np translation ="" prob =""> <np> "* tags, where the *Terminology* based translation and the translation probability are set for each key, respectively.

- Once the input text is provided with the translation options, this text is translated by the SMT system. At the time of translation, apart from the translation option provided by the translation model, the translation options enclosed within XML are considered.

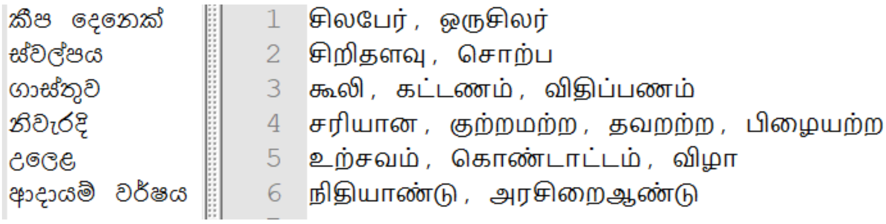


```
<np translation="உதவி தேர்தல் உத்தியோகத்தர்" prob="0.5">
    සහකාර
    <np translation="தேர்தல்" prob="0.5"> මැතිවරණ </np>
    <np translation="ஆணையாளர்" prob="0.5"> කොමසාරිස් </np>
</np>
දිස්ත්‍රික්
<np translation="செயலாளர்" prob="0.5"> ලේකම් </np>
   /
<np translation="அரசாங்க அதிபர்" prob="0.5"> දිසාපති </np>
```

FIGURE 4.15: Sample source text, XML pre-processed based on bilingual translation option.

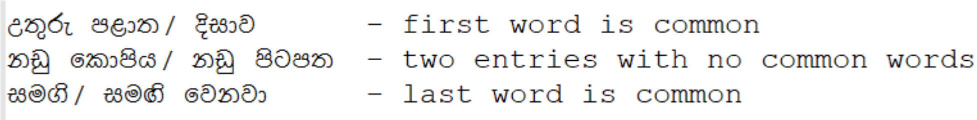
4.4 Dictionary Integration

Apart from the domain-specific bilingual lists (*Terminology*), an open domain dictionary was available for utilization. The entries were extracts of a Sinhala to Tamil dictionary. Therefore, for many Sinhala entries, it had multiple Tamil equivalents (in comma separated form). For each of those equivalents, individual entries were generated (for example, if an entry had three translations for one source word, three new entries were generated). Figure 4.16 illustrates a few such examples. In addition, for some entries, the source side had variations separated with slashes and some of them had common words (tokens) among them. Figure 4.17 depicts a few such entries. In such scenarios, multiple entries were generated based on the original entry. For instance, in case of the first example in figure 4.17, two entries were generated as 'උතුරු පළාත' /uthuru palaatha/ and 'උතුරු දිසාව' /uthuru disaawa/. For the second example, 'තඩු කොපිය / තඩු පිටපත', two entries were generated by partitioning from the slash. For the third example, the generated entries were 'සමගි වෙනවා' /samagi wenawa/ and 'සමඟි වෙනවා' /samagi wenawa/ where the last token is being utilized in both the entries as it is common for both. In cases where there were multiple words in both sides (source as well in the target), all the possible combinations were generated as entries (for example, when the source had 2 variations for which target had 3 variations, 6 new entries were generated).



```
කීප දෙනෙක්      சிலபேர், ஒருசிலர்
ස්වල්පය         சிறிதளவு, சொற்ப
ගාස්තුව          கூலி, கட்டணம், விதிப்பணம்
නිවැරදි          சரியான, குற்றமற்ற, தவறற்ற, பிழையற்ற
උලෙළ           உற்சவம், கொண்டாட்டம், விழா
ආදායම් වර්ෂය     நிதியாண்டு, அரசிறைஆண்டு
```

FIGURE 4.16: Sample dictionary entries that have multiple translation equivalents



```
උතුරු පළාත/ දිසාව        - first word is common
නඩු කොපිය/ නඩු පිටපත   - two entries with no common words
සමගි/ සමඟි වෙනවා       - last word is common
```

FIGURE 4.17: Anomalies in the source side of the dictionary entry that need further pre-processing before utilizing it in the parallel corpus.

This pre-processed list was incorporated into the parallel data similar to *as-corpus*, as described in section 4.3.1.1. Yet the target side entries were not added to the language model as building a language model with single token length entries will not make sense in building a model to learn the sentence flow. This integration is referred to as *open-dic* in the rest of the thesis.

4.5 Name List Integration

In addition to the above mentioned bilingual list, a large list of person names and addresses was available. This list was obtained from a government department and it consisted of names of people and their addresses. The list contained 199,287 entries where approximately half of the entries was names, while the other half was addresses. Each entry had multiple word tokens. The name entries had multiple name tokens (e.g.: first name, last name, family name) while address entries consisted of street names and town.

This list was incorporated into the system in two forms. In the first form, the list was used as is, along with the parallel corpus. No pre-processing was done over the list. The integration technique was the same as what is described in Section 4.4. This integration is named as *name-list* for ease of reference. In the second form, the name list was broken into individual tokens and a list of unique pairs was obtained. Here each entry had only one token. Using human intervention, this list was verified and corrected for translation/transliteration issues. Then this list was incorporated in the same manner as in the first form (*name-list*). The second incorporation is referred to as *name-unique* in the rest of the thesis.

4.6 Handling OOV in SMT

Since the system was built upon very minimal parallel data and the languages are highly inflectional, sparsity in the model was a challenging issue. One of the measures taken to overcome this is paraphrasing, as mentioned in the literature (refer Section 2.4.3).

The languages of interest are low-resourced. However, it was possible to gather a reasonable amount of monolingual data in both the language. Based on these resources, as an avenue to mitigate the OOV, a paraphrase generation technique similar to Chu and Kurohashi [89]'s techniques is proposed.

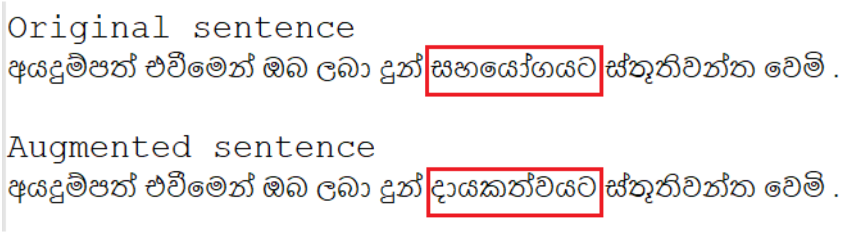
In the approach by Chu and Kurohashi [89], new phrase table entries were created by replacing the occurrences of paraphrases with their corresponding OOV word, along with an extra feature to distinguish these new entries. Also, they used the retrofitting concept of Faruqui et al. [94], to retrofit semantic lexicon to achieve better quality on paraphrasing. This approach gave promising results for English, which has many good semantic lexicons (Word-Net [91], FrameNet [92], and the Paraphrase Database (PPDB) [93]). However, English is less inflected than Sinhala or Tamil, which have the possibility of the existence of morphological variations. Also, Sinhala does not have any well performing semantic lexicons. Moreover, inflections play a major role in creating OOVs (e.g., The word 'ගස' /gasa/ ‘tree’ gets transformed into a different word based on inflection as 'ගසක්' /gasak/, whereas in English, this simply becomes ‘a tree’, without any change in the base word). This agglutinative nature of the language causes the system to treat the inflections as different words and lead to high rate of OOV.

Having considered the above limitations and available resources, in this thesis, four heuristics are proposed to retrofit the output of the word embedding in order to give precedence to inflections and typographical related issues (mistakes made during typing such as spelling mistakes). However, due to the lack of semantic lexicons, this will not be like the semantic-based filtration done using the semantic lexicons as in Chu and Kurohashi [89]’s implementation. Also, they evaluate the impact of the paraphrasing techniques proposed only by augmenting the phrase table. We evaluate the impact of paraphrasing techniques by using another two more integration approaches apart from augmenting the phase table.

In the first approach, the parallel corpus is modified by adding a synthetic parallel corpus that is generated from existing parallel data by replacing the paraphrases via its corresponding OOV word. For example, the word ‘දායකත්වයට’ /daayakathwayata/ (for the assistance) is not found in the parallel data. Yet ‘සහයෝගයට’ /sahayoogayata/ is suggested as a paraphrase to it. The occurrence of these words in the parallel data (in the source side) is found. Then the new sentence is generated by replacing the word ‘සහයෝගයට’ by ‘දායකත්වයට’. New parallel data is generated through this (the target side sentence is used with no changes). Figure 4.18 shows an original sentence and the corresponding augmented sentence for this example. Here OOVs in test and tuning with respect to training data are considered. This is referred to as *re-train*.

In the second approach, the procedure of Chu and Kurohashi [89] is followed. This is referred to as *re-tune*.

In the third approach, no changes are made to the data, the model, or the feature weights. Instead, test data is pre-processed before sending for decoding. Thus, before decoding, OOVs are identified and paraphrases for OOVs are fetched as in *re-train* and *re-tune*.



Original sentence
අයදුම්පත් එවීමෙන් ඔබ ලබා දුන් සහයෝගයට ස්තූතිවන්ත වෙමි .

Augmented sentence
අයදුම්පත් එවීමෙන් ඔබ ලබා දුන් දායකත්වයට ස්තූතිවන්ත වෙමි .

FIGURE 4.18: An example original sentence and an augmented sentence. The paraphrase and the OOV are highlighted.

Then the translations for these lists of words are retrieved from the same translation engine. This pre-information on the translation equivalents is used as an extra piece of information for the decoding process by providing those translations as annotated inputs along with the similarity value as the translation probability. This is referred to as *pre-process*.

For all three techniques (*re-train, re-tune* and *pre-process*) above, multiple experiments were carried out based on the number of paraphrases (the number of words) that should be taken into consideration. For each OOV, the 10 most similar words were captured from the embedding model, if that OOV word exists in the word embedding model. This list was filtered based on three different criteria:

- Considering different numbers of paraphrase options for each OOV (number of words considered: 10, 5, 2 and 1 where the priority is given based on the similarity value).
- Considering paraphrases with the similarity value above a given threshold (threshold values considered were 0.9, 0.8, 0.7 and 0.6).
- Morphological heuristic-based filtering over the paraphrases retrieved from the word embedding model, with the motive to refine the list to find the most suitable paraphrases for the OOV by considering inflection variations.

  The following are the different heuristic-based filtrations used to retrofit the result returned from the embedding model:

- *stem:* Filter to identify the inflectional variations of OOV.
  E.g., For the word 'කලාපයන්හි' /kalapayanhi/ - (in zones), it considers the words starting with 'කලාපය' /kalapaya/ -‘zone’, which is the base word. (In Sinhala, inflections are added as suffixes to the base form. Here we consider whether a candidate word has the starting pattern as of the OOV in order to find the inflection variants).

- *inflection:* Filter to identify the words with the same inflection as the OOV.
  E.g., For the word 'සමඟියෙන්' /samagiyen/ - (in unity), it picks the word 'සහෝදරත්වයෙන්' /sahodarathwayen/- (in brotherhood) - a similar word with the same inflection.

- *similarity:* Filter to identify words with minor spelling variations (e.g., typographical errors).
  E.g., 'සෙවණ’ and 'සෙවන' are both pronounced as ‘sevana’ and are used to mean ‘shelter’ though the first one has the correct spelling. However, in the writings the occurrence of both the words are noticed.

- *combined:* This combines the results of all above 3 filtration techniques. It identifies synonyms with similar inflectional variations, different inflections of the same word, and words with slight spelling variations.

4.7 Summary

This chapter described the techniques proposed to improve the translation quality of the Si-Ta baseline system.

They cover domain adaptation, terminology integration, dictionary integration, name list integration, and OOV handling. A novel data augmentation technique was presented under the terminology integration section where it was utilized to generate new parallel data out of the parallel corpus and a bilingual list.

In addition, a paraphrasing technique based on novel heuristic-based filtering was also presented.

The next chapter will report the results of each of the techniques presented here.

Chapter 5

Evaluation and Analysis

5.1 Introduction

This chapter presents the evaluation results and analysis of the techniques presented in the thesis so far. The set of experiments done under each sub-topic are divided into subsections. They are: 1. Baseline system, 2. Domain adaptation, 3. Terminology integration, 4. Dictionary integration, 5. Name list integration, and 6. Out-of-vocabulary handling.

The performance of the system in all the experimental setups is evaluated using BLEU [53]. OOVs are reported for some experiments to highlight the influence of the enhancements on reducing the OOV. The same set of 300 random unique sentences is used in all the evaluations. The test set includes sample sentences from official letters of different government organizations.

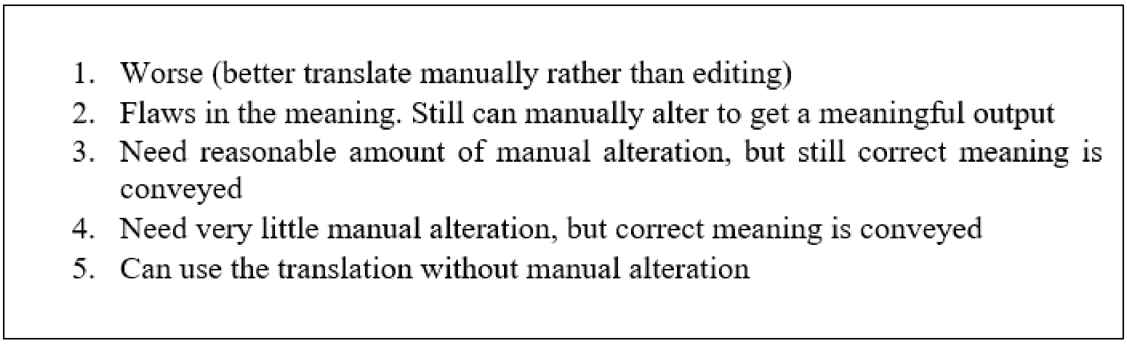
5.2 Baseline

The baseline system was configured as per the description given in Chapter 3. The system was set up and evaluated in both Si-Ta and Ta-Si directions for the same training, tuning and testing sets. Yet the monolingual data that was used to build the language model was not the same, as single side data was gathered from different sources (see

Tables 3.1, 3.2 and 3.3). The evaluation scores (in BLEU), number of OOVs, along with the BLEU scores for the same test set for the output produced by the Google Translate[1] are reported in table 5.1.

Apart from the automated evaluation, two forms of human evaluations were carried out to evaluate the system usability.

In one form of evaluation, the outputs of Si-Ta, as well as Google, were evaluated based on a rating scheme. The rating was carried out on a 5-point Likert scale (refer to Figure 5.1). Twenty seven human translators took part in this evaluation. They were randomly split into nine groups where each had three translators. Ten source sentences along with their translation outputs from either of the systems were given to each group. The translators were requested to score the output based on the translation quality. But the system of origin of each translation was not indicated. As the participants were fluent only in Sinhala-Tamil direction, Tamil-Sinhala direction was not evaluated. From the highest score of 5, our Si-Ta system earned 3.2 and Google Translate earned 2.4.



1. Worse (better translate manually rather than editing)
2. Flaws in the meaning. Still can manually alter to get a meaningful output
3. Need reasonable amount of manual alteration, but still correct meaning is conveyed
4. Need very little manual alteration, but correct meaning is conveyed
5. Can use the translation without manual alteration

Figure 5.1: Five point Likert scale used to evaluate the translation outputs of Si-Ta system and Google translate.

[1] https://translate.google.com/

Table 5.1: Evaluation scores for the baseline system

Direction	System(BLEU)	Google(BLEU)	System(OOV)
Si→Ta	24.63	8.05	286
Ta→Si	32.04	8.37	342

Next evaluation was performed to identify if any improvement is attained in the performance in the translation process when Si-Ta system is incorporated. For this evaluation, 4 translators who were familiar with our system were employed. Since they were conversant only in Sinhala-Tamil direction, the experiments were carried only in that direction. For the evaluation, 4 approximately equal sized (letters of approximately 100 words) government official letters were picked. The translators were instructed to translate two of the letters manually, and post-edit the output of Si-Ta system to get the translation for rest of the letters. Time required for the complete translation for each letter was tracked.

The time required for the translation along with the average required time for each translation type (manual, using the system) and average time variation between each type are tabulated in Table 5.2. Translation based on Si-Ta post-editing and manual translation are referred to as *S<n>* and *M<n>* respectively, where *n* refers to the document number. The *Avg* refers to the average time required for each kind of translation, while *Avg Diff* refers to the different between the average time required for manual translation and average time required for translation based on Si-Ta post-editing.

Analysis

In general, in either directions, the BLEU scores for Si-Ta system are noticeably higher with respect to the scores of Google Translate. Similarly, the scores are higher than the previously reported work [4, 58]. This result can be related to the data type used to train these systems. Si-Ta is exclusively trained with official documents as parallel data, while the other systems were trained with open domain data. Language flow as well as

TABLE 5.2: Time taken (in minutes) for translation of text manually and by post-editing the Si-Ta output for each translator

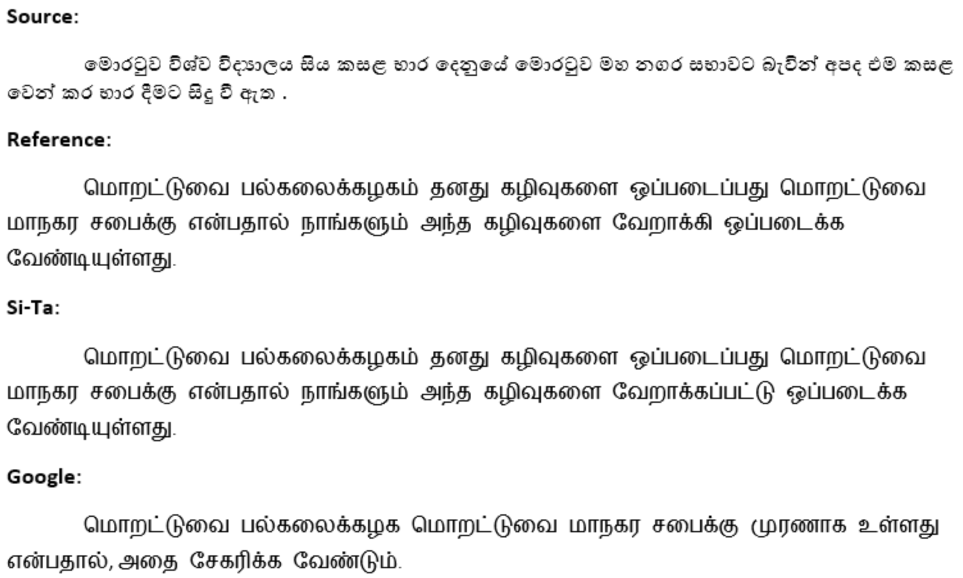
Translator	S1	S2	Avg	M1	M2	Avg	Avg Diff
T1	11	5	8	8	11	9.5	1.5
T2	9	8	8.5	9	12	10.5	2
T3	3	4	3.5	5	8	6.5	3
T4	14	12	13	18	20	19	6

the vocabulary, show less divergence when it comes to official documents compared to the open domain, also the writing style is formal. These scores reveal the prominence of domain adaptation at the time of implementing a MT systems for different domains such as legal, government, and medical. In order to achieve good results, domain-specific corpus is required [70].

Apart from that, the well performed translation direction was Tamil-Sinhala. The logical reasoning is that as Tamil is more inflected than Sinhala therefore, the complexity of Tamil-Sinhala direction is lesser compared to Sinhala-Tamil. (Si-Ta direction has one-many while Ta-Si direction has many-one translation). This has been proven in previous experiments [56–58] as well. This same reason causes a higher number of OOVs in Ta-Si direction than in the reverse direction.

Manual analysis of Google Translate output revealed that its translated output was acceptable for phrases, but not for full sentences. In many scenarios, for lengthy sentences, the meaning of translations were much deviating from that of the source sentence. Figure 5.2 highlights one such scenario. Here the meaning of the original sentence is: ”As the University of Moratuwa handovers the garbage to the Moratuwa Municipal Council; we are also required to separate that garbage and handover.”. The output of the Si-Ta system expresses the same meaning but with a minor inflection variation which can be easily post-edited. However, the output of Google Translate means: ”As the University of Moratuwa is against Moratuwa Municipal council; need to collect it.”, which is completely different information from that of the source sentence. Further, some had literally correct translations, which were not context appropriate. In addition, there were occasions when system failed to translate and gave a blank output.

Furthermore, manual evaluation scores and time consumption for post-editing over manual translation reveal that Si-Ta – at its current stance has the capability of giving a meaningful translation where it can be utilized with post-editing to speedup the manual translation process.



Source:

මොරටුව විශ්ව විද්‍යාලය සිය කසළ භාර දෙනුයේ මොරටුව මහ නගර සභාවට බැවින් අපද එම කසළ වෙන් කර භාර දීමට සිදු වී ඇත .

Reference:

மொறட்டுவை பல்கலைக்கழகம் தனது கழிவுகளை ஒப்படைப்பது மொறட்டுவை மாநகர சபைக்கு என்பதால் நாங்களும் அந்த கழிவுகளை வேறாக்கி ஒப்படைக்க வேண்டியுள்ளது.

Si-Ta:

மொறட்டுவை பல்கலைக்கழகம் தனது கழிவுகளை ஒப்படைப்பது மொறட்டுவை மாநகர சபைக்கு என்பதால் நாங்களும் அந்த கழிவுகளை வேறாக்கப்பட்டு ஒப்படைக்க வேண்டியுள்ளது.

Google:

மொறட்டுவை பல்கலைக்கழக மொறட்டுவை மாநகர சபைக்கு முரணாக உள்ளது என்பதால், அதை சேகரிக்க வேண்டும்.

FIGURE 5.2: An example where Google Translate gives a misleading translation.

5.3 Domain Adaptation

With the motive to achieve more fluency in the output, experiments were carried out to evaluate the domain adaptation techniques explained in section 4.2. Results for multi-models: log-linear (*log*), linear (*linear*) and log-linear with filtered out-domain (*filter*), in either direction are reported in Table 5.3. Number of OOVs are reported since there are no changes in the OOVs across the setups as the impact is only on the language model.

Analysis

As reported in Table 5.3, when data is split (based on its relevance to the domain) and individual models are created and utilized, improvements in the scores are observed.

TABLE 5.3: Translation score variations for different language model configurations.

Setup	Si→Ta	Ta→Si
baseline	24.63	32.04
log	24.91	32.41
linear	24.78	32.36
filter	24.88	32.31

Yet no changes were made to the translation model. The factor that affects the translation quality here is the language flow. Selection is differentiated by the priority given to different data sources, for instance, out of the available translation options, which translation will suit the most in the given context. Special consideration is given for the selection of the most appropriate inflection variants based on the context. For example, the word 'අවසන් වසර' (/awasan wasara/) in Sinhala means 'final year'. However, the word 'අවසන්' (/awasan/) alone can mean 'last', 'final', 'ending', or 'finish'. In the baseline system, the system translated this Sinhala phrase into the Tamil phrase 'நிறைவு ஆண்டு'(/niraiwu aandu/), where the word 'நிறைவு' (/niraiwu/) bears the meaning 'ending' or 'finish', which is not appropriate. When multiple language models are used in log-linear fashion, this got translated into the correct term as 'இறுதி ஆண்டு' /(iruthi aandu/), where the meaning is 'final year'.

In addition, the evaluation score for the log-linear language models was marginally higher than that of the linear-based models. The logic might be that in log-linear models, the weight adjustment is done based on the translation scores while in linear models the priority is set based on the perplexity value of the tuning set. Yet in literature, both are recommended over the single models, while no comparison has been made between them [61].

However, using the same data in filtered mode did not show any improvement over the scores, though the perplexity of the filtered data was lower than that of unfiltered data (refer tables 4.1 and 4.2 for perplexity values).

One reason given by Moore and Lewis [70] is that, as the perplexity reduces, more weight is given to the out-domain LM, though still, the writing style is drastically different. This can be the reason, even in our experiments as well.

For example, the phrase 'சிக்கலான நிலையை உருவாக்கியுள்ளது'. (/sikkalaana nilaiai uruwahiyulathu/) means 'has created a problematic situation', where the word 'சிக்கலான' (/sikkalaana/) is meant to be 'problematic' though it can also take the meaning as 'complex', 'issue', or 'conflict'. Without filtered out-domain data, the phrase is translated as 'ගැටළු සහගත

වී ඇත' (/gatalu sahagatha wee atha/), which is the correct translation. However, when the out-domain data is added after filtration, the system translates it as 'සංකීර්ණ වී ඇත' (/sankeerna wee atha/), which means 'has become complex' which is not the proper translation for the given context.

The perplexity value of the Tamil out-domain corpus (2210.4860) is comparatively higher than that of Sinhala out-domain corpus (918.3833). Tamil out-domain corpus had more blog articles where the writing style was more informal with more colloquial style text. Sinhala out-domain corpus had more news articles where the writing style is less colloquial. This could be the reason for this high variation in perplexity values.

The experimental results reveal that though it is recommended to have a larger amount of data to build a language model to have better coverage, the quality and the relevancy of the data plays a vital role. Therefore, it is important to differentiate the priority based on applicability.

5.4 Terminology Integration

The experimental results reported herewith are for the techniques explained under section 4.3. As the prime goal was Sinhala-Tamil direction, experiments were carried out only in that direction. The evaluation results are reported in two sections for clarity. The integration techniques other than the ones with data augmentation are reported in tables 5.4 and 5.5, while the scores related to the data augmentation techniques are reported in Table 5.6, as the augmentation was carried on with only one list type (*TERM-1*).

Analysis

When the list is utilized in the corpus along with the parallel data, and trained (*as corpus*), the effect of data is twofold. First, the gaps in the vocabulary is filled, second, more the data, the better the word alignment heuristics [2]. This impact was noticeable when observing the word alignments outputs of each experimental setup. There were

Table 5.4: BLEU scores for different term integration techniques for bilingual term integration (the higher the better).

	as corpus	*multiple-table*	*merge-table*	*dynamic*
baseline	24.91			
TERM-1	25.31	24.95	24.94	24.62
TERM-2	25.17	24.89	24.92	24.23

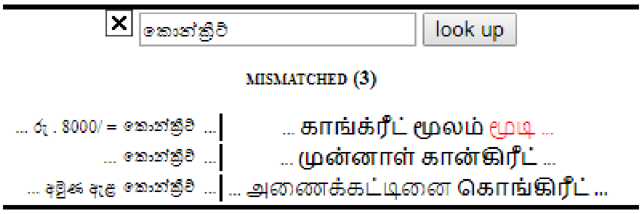
Table 5.5: Number of OOVs for different term integration techniques for bilingual term integration (the lower the better)

	as corpus	*multiple-table*	*merge-table*	*dynamic*
baseline	286			
TERM-1	243	263	265	232
TERM-2	235	259	263	226

Table 5.6: Evaluation scores for integration of augmented parallel data generated based on different techniques.

	BLEU	OOV
baseline	24.91	286
'Based on ending word'	25.32	249
'Based on ending word + POS'	25.46	241
'Based on ending word + improved POS'	25.67	232
Based on NER	25.38	246

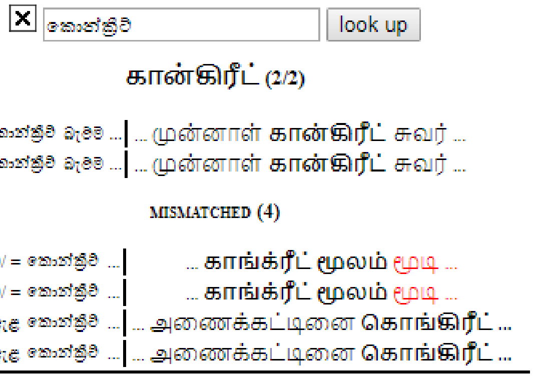
scenarios where the words that were not translated in baseline experiments as a result of misalignment, were translated in the latter, since they were aligned accordingly. Figures 5.3 and 5.4 depict the screenshots of word-alignment output, which reveals this.



MISMATCHED (3)

... රු . 8000/ = කොන්ක්‍රීට් ...	... காங்க்ரீட் மூலம் மூடி ...
... කොන්ක්‍රීට් ...	... முன்னாள் கான்கிரீட் ...
... අලුත් ඇළ කොන්ක්‍රීට් ...	... அணைக்கட்டினை கொங்கிரீட் ...

Figure 5.3: Alignment for the Sinhala word 'කොන්ක්‍රීට්' /concrete/ (concrete) in the baseline system (Although the equivalent Tamil word '**கான்கிரீட்**' /concrete/ exists, it is misaligned).



☒ කොන්ක්‍රීට් look up

காண்கிரீட் (2/2)

... කොන්ක්‍රීට් බැම්ම ... | ... முன்னாள் கான்கிரீட் சுவர் ...
... කොන්ක්‍රීට් බැම්ම ... | ... முன்னாள் கான்கிரீட் சுவர் ...

MISMATCHED (4)

... රු . 8000/ = කොන්ක්‍රීට් ... | ... காங்க்ரீட் மூலம் மூடி ...
... රු . 8000/ = කොන්ක්‍රීට් ... | ... காங்க்ரீட் மூலம் மூடி ...
... අමුණ ඇළ කොන්ක්‍රීට් ... | ... அணைக்கட்டினை கொங்கிரீட் ...
... අමුණ ඇළ කොන්ක්‍රීට් ... | ... அணைக்கட்டினை கொங்கிரீட் ...

FIGURE 5.4: Alignment information for the same example as in figure 5.3, for 'as corpus' setup. Here the equivalent Tamil word is aligned.

In *multiple tables*, in addition to the primary phrase table, separate phrase tables are generated from *Terminology*. Entries in *Terminology* were either single words or short phrases. Further, there were multiple occurrences of the same source words with distinct words in the target side (in *TERM-2*). This influence the word alignment heuristics in picking up adequate statistics. Moreover, there were short phrases or single word entries in this phrase table. This leads to poor quality phrase table [2] (for a (source) word with multiple entries (distinctive target words), equal probability is set for all, as not further information is available about the context within the list. This confuses the selection of most desired mapping).

Nonetheless, the weights of these phrase tables are set according to their influence on the tuning set. It negates the above mentioned adverse effect. Yet, still this may create confusion for the entries with equivalent source with distinct probabilities. This can be the reasoning for the slight reduction in the BLEU. For example, in *TERM-2*, there were higher number of multiple entries (for the same individual source entry). Also, it had single word entries/short phrases in higher numbers. For instance, for the Sinhala word 'ආදර්ශ' /aadarsha/ - the lexical probability for the word '**மாதிரி**' /maathiri/ (model) was 0.333303 in the primary phrase table, while it was 0.289698 in the phrase table generated from *TERM-2*. Yet, this second phrase table had many other translations that did not occur in the primary table, for instance '**முன்மாதிரி**' /munmathiri/ (prototype)) and ('**குறிக்கோள்**', /kurrikkol/ (objective). Thus, if the word 'ආදර්ශ' is found in the tuning data with given reference as ‘மாதிரி’, the priority given to the second phrase table will be lesser. If the reference had the translation as one of the options provided in the secondary phrase table, a higher weight will be assigned to that secondary table. This sort of situations will create confusion in providing appropriate weights.

However, in *merged tables*, the above mentioned confusion is mitigated as the duplicate entries (same translation, but with different statistics) are dropped since the tables are merged. Also, a new feature is added to the merged table to distinguish the secondary table entries. Although there is no improvement in the scores, it is not declined, as in the previous technique. Here, for the example explained in the previous techniques (*multiple tables*), the probability is set to 0.333303 which is taken from the primary table.

In *dynamic*, before the test set is processed by the decoder, it is annotated for the word/phrase that exists in *Terminology*. This integration techniques shows drop in scores for both the lists. There are many factors for this drop. First, as no changes done to the weight or the model, the advantageous/adverse effect is unpredictable. Hence, optimizing/managing the effect is less practical. Second, at the time of annotation, the word/words that match an entry in the list is considered to be the boundary. Yet, there are many situations when a shorter span is annotated while the precise span is longer than what is being annotated, also the translation model (phrase table) had the translation for the longer phrase. Since this misguide the decoder, it makes an impact on the translation quality. When the list had more short entries, the negative impact was higher. For example, the Sinhala term ‘ශ්‍රී ලංකා ගුවන් විදුලි සංස්ථාව’ /Sri lanka guwan widuli sangstawa/ (Sri Lanka Broadcasting Corporation) is annotated as follows:

‘ශ්‍රී ලංකා ගුවන් <np translation = ”மின்சாரம்” prob=”0.5”> විදුලි </np> <np translation=”கூட்டுத்தாபனம்” prob=”0.5”> සංස්ථාව </np>’.

Here the compound word ‘ගුවන් විදුලි’ /Guwan widuli/ (broadcast) is split based on the annotation, Therefore, the word ‘විදුලි’ /widuli/ is annotated with the word ‘மின்சாரம்’

/minsaaram/ (electricity) with a higher probability. This is the reason for getting a lower BLEU score (worse) though a lower OOV (better) is noted for *dynamic* integration.

Yet, in the prior work [81], this integration technique has shown improvement in the scores. However, these experiments were done with the parallel data from open domain, and the lists were domain specific. In addition, in prior research, ratio of entries in the list was smaller when compared to the number of sentences in the parallel data. Yet, this ratio was much higher in our system (*TERM-2* – 19,861 list entries while parallel data - 24,817 sentences). Also, the implementation of dynamic integration was simple, especially the implementation of the term identification step. It is implemented based on basic look-up with no consideration given to the inflection variations and the term boundary detection was not strong. Improvements in these areas may contribute towards improvement in translation as well.

Collectively, by carefully analyzing the translation output of the configurations which showed improvement in scores, it is apparent that many of untranslated words were in their inflected forms. Yet, the list had only the base forms of those untranslated words. Here the languages of concern are highly inflected. And each inflectional form is treated as a distinct word. Hence, if the bilingual list consists the inflectional variations, it will support in improving the translation quality.

Experimental results for the data augmentation techniques explained in the section 4.3.1.4 are tabulated in Table 5.6. As reported, all augmentation techniques showed improvement over the baseline, while *'Based on ending word + improved POS'* showed the best result based on BLEU as well as on OOV. This improvement is mainly due to our heuristic to reduce the influence of the mismatching in inflection between Sinhala and Tamil (Tamil is more inflectional). In all the other techniques other than *'Based on ending word + improved POS'*, the Sinhala candidate term is identified, the corresponding Tamil translation is identified, and then no further analysis is done on the inflection variation. When the corresponding Tamil word carries the inflection within itself, while Sinhala word is in base form with the preposition as a separate token, the Sinhala preposition is left alone in new augmented parallel data. For example, the phrase ‘පර්යේෂණ අංශය මගින්’ /paryeshana anshaya magin/ (by the research unit) is translated as ‘ஆராய்ச்சி பிரிவினால்’ /aaraichchi pirivinaal/ in Tamil where the preposition ‘by’ is agglomerated to the word ‘பிரிவு’/piriwu/ (unit). Using this kind of sentences for data augmentation will mislead the alignment. *’Based on ending word + improved POS’* is a novel contribution from us, where the adverse effect of such inflection mismatch is reduced to a certain extent while giving context to the term.

5.5 Dictionary Integration

This section reports the evaluation scores for the integration of a general purpose (open domain) dictionary to the system as described in section 4.4. The evaluation scores are as reported in Table 5.7.

Analysis

This integration reports a slight improvement in the BLEU score and a considerable reduction in the number of OOV. The main factor behind it is, most dictionary entries were single words. This impacted negatively for multiple-source-single-target (many-to-one) form translations.

For instance, many Sinhala compound phrases (multi-word) were translated to a single word in Tamil, while each of the individual words within the compound word had its own equivalent Tamil translations. (The Sinhala sentences were 14 word long - in average, while in Tamil, it was 12). When integrating lists that have a high number of short length entries, the system favours word-by-word translation even for compound

Table 5.7: Evaluation scores for (open domain) dictionary integration

	BLEU	OOV
baseline	24.91	286
open-dic	25.18	217

words. Also, the integration misleads the alignment of compound words, this leads to incorrect output.

For instance, consider the following scenario,
Source: ක්‍රියාත්මක වේද යන්න /kriyathmaka wedha yanna/
Machine Translation: இயங்குகின்ற வேதங்கள் என்பதை /iyaguhindra wedangal enpathai/
Reference: செயற்படுத்தப்படுகின்றதா என்பதை /seyalpaduththapakindratha enpathai/

For the above mention instance, the source ‘ක්‍රියාත්මක වේද’ /kriyathmaka wedha/ refers to - “whether it works”, the corresponding Tamil translation for this compound Sinhala word is ‘செயற்படுத்தப்படுகின்றதா’ /seyatpaduththappadukiratha/. However, with the dictionary integrated, the word ‘වේද’ /wedha/ is translated to ‘வேதங்கள்’ /wedhangal/, where the meaning is “religion”. However, this is incorrect for the context given though, it is a valid translation when only the word ‘වේද’ is considered.

5.6 Name List Integration

The evaluation scores for integration of a bilingual name list described in section 4.5 are tabulated in Table 5.8.

Analysis

This integration has produced an increase in the BLEU score, as well as a reduction in the OOV count as the official letters had names of people as well as places in many instances (the headers and footers had names and addresses). The *name-list* had better

TABLE 5.8: Evaluation scores for integration of a bilingual name list

	BLEU	OOV
baseline	24.91	286
name-list	26.26	212
name-unique	25.23	226

improvement in the score than the *name-unique*. Manual analysis showed that *name-list* had a better word alignment. In *name-list,* having multiple occurrences in different contexts while in *name-unique* had only a single occurrence as a single entry may be the reason for this variation in alignment.

However, a negative effect of this integration was observed as well. There were instances where the words were getting transliterated when translation is required. For example 'එය අළුත් කර ගැනීමට ' /eya aluth karaganeemata/ (to renew it) was translated as '**அலுத் செய்து கொள்வதற்கு**' /aluth seithu kolwathatku/ which should be translated as '**புதுப்பித்துக் கொள்ள**'/puthuppiththu kolla/. Here the word 'අළුත්' /aluth (new) is being transliterated instead of translated. This happens since the list contained the transliteration equivalents of common words instead of translation when they appear in person names and place names. However, in a general context, considering this transliteration will lead to incorrect output.

5.7 Handling Out-of-Vocabulary Words

As per the approaches discussed in Section 4.6, the impact of embedding-based paraphrasing on OOV reduction was evaluated. Evaluations were done in Sinhala-Tamil direction. The performance was analyzed for three different setups (*re-train, re-tune* and *pre-process*) for 12 different filtration techniques (4 experiments based on the number of paraphrases, 4 experiments based on the threshold value of similarity metric, 4 experiments based on filtration using simple heuristic-based morphological filtration).

Evaluation scores are tabulated in Table 5.9. Each row represents the evaluation results based on a single filtration technique. 'baseline' denotes the system with no paraphrasing, the next four rows denote the number of considered paraphrases, respectively 1, 2, 5 and 10. Next, the results for the threshold based filtration are shown, where the threshold values considered were 0.9, 0.8, 0.7 and 0.6. The last four rows denote the heuristic filtration techniques discussed in Section 4.6. For these 8 experiments, the columns are named in 'n-m' convention, where n –denotes the mode (0: considering the number of words for paraphrasing, 1: Considering a threshold value over the similarity), m- denotes the value of consideration. Table 5.10 presents the OOV counts, and the order of the rows is the same as in Table 5.9.

Based on the BLEU scores (Table 5.9), on average, there is a gain for each setup. The highest gain is achieved with respect to the baseline were 0.93, 0.58, and 0.41, respectively, for *re-train, re-tune* and *pre-process*.

The highest gain of 0.93 was reported in *re-train*. The impact is twofold. They are: the coverage over OOV, and more parallel data leads to better alignment [2].

For *re-tune*, an alteration was done on the phrase table by amending new translation entries for the OOVs, yet this did not make any changes for existing translation entries (unlike in *re-train*). Therefore, consideration of multiple number of options of paraphrases (1, 2, 5 and 10) for each OOV did not show any difference in the OOV rates. Yet, there is a fluctuation in the BLEU scores. Since multiple translation options are given, there will be a variation in the translated output.

In *pre-process*, the impact of paraphrases is imposed to the system only at the time of decoding. No improvements were made over the training data or the model (translation

Table 5.9: BLEU scores for word embedding-based paraphrasing over OOVs

	re-train	re-tune	pre-process
baseline	24.91		
0-1	25.00	25.29	25.16
0-2	24.93	25.32	25.05
0-5	25.17	25.30	24.95
0-10	25.48	25.40	25.20
1-0.9	24.89	25.04	24.92
1-0.8	24.97	24.90	24.90
1-0.7	25.63	25.25	25.11
1-0.6	25.55	25.23	25.04
stem	**25.84**	**25.49**	25.12
inflection	25.28	24.93	25.09
similarity	25.26	25.06	25.19
combined	25.64	25.11	**25.32**

model), or the feature weights. Therefore, the impact on the score is less prominent than that of the other two setups. However, the practical significance of this method is higher in a production workflow as this procedure addresses the issue in a dynamic context.

When considering the overall results of the paraphrasing techniques, a few points can be highlighted. For many words, the system was able to predict the synonyms as the paraphrases. This resulted in the correct/meaningful translation for OOV words.

For the inflected word forms, the system provided both the base form and the other inflected variations. This leads the translation to be an inflected variation of the desired output. Therefore, a human translator who is conversant only in the target language can correct and make use of this translation with less effort than it is being untranslated.

Another OOV present in this test data is due to spelling variations (the same word is written using different spelling due to human error). It was possible to get the word with the correct spelling as a paraphrasing option from the embedding model. For example, the word ‘කිහිපදෙනකු’ /kihipadeneku/ was written as ‘කීපදෙනෙකු’ /keepadeneku/ (they both mean the same –‘few people’) in the input test data. This left untranslated in

TABLE 5.10: OOV counts for use of word embedding-based paraphrasing over OOVs

	re-train	re-tune	pre-process
baseline	286		
0-1	221	159	165
0-2	207	159	163
0-5	188	156	163
0-10	163	159	156
1-0.9	233	241	241
1-0.8	219	216	231
1-0.7	187	186	228
1-0.6	170	164	225
stem	199	207	208
inflection	223	210	211
similarity	202	201	201
combined	189	185	185

the baseline system, since the system did not have the word ‘කීපදෙනෙකු’ in the translation model. However, the word ‘කිහිපදෙනකු’ was identified as a paraphrase by the embedding model and translation equivalent was found. The other type of OOV was Named Entities (NEs). The system returned NEs of the same category as paraphrases (for a female name, a list of female names). This leads the translation to be incorrect, which is a negative impact of using word embedding. Apart from that, filtration based on the heuristics has shown a positive BLEU score gain over the techniques without filtration.

5.8 Summary

This chapter reported on the evaluation results for the experiments that were carried out along with the reasoning for the variation in the scores. The results for each technique were compared with those of the baseline system and outputs were analyzed to identify the causes for the variations. The results reveal that all the experimented techniques other than data filtration gave scores above the baseline.

Chapter 6

Conclusion and Future Work

In this thesis, we presented the first translation system with domain adaptation for Sinhala-Tamil language pair. The languages are low-resourced and highly inflected. The domain of consideration is official government documents. The prime idea behind this research is to develop an automated MT system to be used as an intermediate step in the translation workflow. The idea is to accelerate the human translators' work by post-editing the output of the MT system. The evaluation results reveal that hypothesis on reaching the goal is attained.

As the initial step, the focus was given to build a baseline SMT with the available data. Selection of SMT over the other MT implementation methodologies was based on the practical feasibility of its implementation with minimum groundwork. The evaluation scores were far better than the scores for Google Translate.

The main focus of this thesis was to identify avenues to improve the baseline system. We focused on three main areas to improve the translation quality, considering the available resources and limitations. They are: 1) Domain adaptation, 2) Terminology integration and 3) Handling OOV. Experiments were conducted to evaluate the impact of each technique proposed based on BLEU scores and the number of OOVs. The results and analysis were reported.

Under domain adaptation, multiple language models were created and different integration techniques were evaluated. The evaluation scores revealed that having multi-models either in linear or log-linear, performed better than having a single model. However, the data filtration over the out-domain data based on perplexity did not show a positive move, though there was improvement in the perplexity values.

A novel data augmentation technique was presented under terminology integration. Use of this augmented data in training the SMT system showed a promising result than the list as bilingual entries. This data generation technique is fairly general for any language pair which lacks a reasonable amount of parallel data.

Apart from that, an open domain bilingual dictionary, and a bilingual name list consists of person names and addresses were integrated and evaluated. The name list gave promising improvement while the dictionary did not contribute significantly.

Word embedding based paraphrase techniques were experimented under OOV handling, where a novel heuristic filtering showed a better result in retrofitting. The evaluation scores and manual analysis reveal that this approach is beneficial. The proposed novel technique is applicable for most of the Indic languages as they are highly inflectional, and mostly the inflection is added as a suffix to the base word [115]. This system can be deployed for a computer-aided translation workflow in a low resourced setup. It can be used to ease the workload of post editing, by giving better suggestions for the OOVs rather than leaving them untranslated.

Yet, the implemented system and the techniques experimented do have limitations. They are listed below along with the possible improvement points:

- Currently, the developed SMT system does not have a special capability to handle the inflections. Integrating techniques such as factored models [116] will help to improve the translations of inflections.

- Though the domain for this research is considered to be government official documents, experimental results reveal that context differs from organization to organization. Therefore, for the system to function in a more effective way, the system needs more fine-tuning in domain adaptation. This requires dynamic domain adaptation, similar to the idea proposed by Foster and Kuhn [65]. This will require more work on domain detection.
- The terminology lists used for experiments with terminology integration were in canonical form while both the languages are highly inflected. Generating the inflections in both the languages and populating the parallel list would be a valuable resource addition.
- A shortcoming with the proposed paraphrase based OOV handling method is that it mishandles the named entities. To eliminate this issue, a good NER system needs to be integrated into this workflow to detect the NEs beforehand and to handle them through a transliteration module.
- The key drawback in the system is, it is running with a very low-resource while the system requires quality data in a larger amount. Increasing the parallel corpus is an ongoing process.

In addition, following action points can be considered as useful future work in improving the system as an end product, since the prime goal is to develop a translation system to assist the human translators.

- Since the output of the system is supposed to be post-edited by the human to get the final output, incorporating a robust post editing framework such as 'casmacat workbench' [117] will make the system much user friendly.
- Since the translation outputs are being corrected and verified by human translators on a daily basis, the system receives new data. Yet, the correction based on live feedback is not shown-up in the future translation tasks without re-training

the models. Therefore, incorporating methods to facilitate dynamic phrase tables such as suffix array [118] will be worthwhile.

- Based on the work under OOV handling, it is clear that the paraphrase suggestion based on word embedding is fairly acceptable. Use of this concept, in post-editing work to provide potential suggestions, is a viable option to support post-editing.

Bibliography

[1] Bernard Vauquois. A survey of formal grammars and algorithms for recognition and transformation in mechanical translation. In *Ifip congress (2)*, volume 68, pages 1114–1122, 1968.

[2] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[3] Fathima Farhath, Pranavan Theivendiram, Surangika Ranathunga, and Sanath Jayasena. Improving Domain-specific SMT for Low-resourced Languages using Data from Different Domains. In *Eleventh International Conference on Language Resources and Evaluation(LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018.

[4] Randil Pushpananda, Ruvan Weerasinghe, and Mahesan Niranjan. Statistical machine translation from and into morphologically rich and low resourced languages. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 545–556, 2015.

[5] Doug Arnold. *Machine translation: an introductory guide*. Blackwell Pub, 1994.

[6] Arturo Trujillo. *Translation engines: techniques for machine translation*. Springer Science & Business Media, 2012.

[7] John Hutchins. Machine translation: History and general principles. *The encyclopedia of languages and linguistics*, 5:2322–2332, 1994.

[8] Dan Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2009. ISSN 08912017.

[9] Eric H Nyberg and Teruko Mitamura. The kant system: Fast, accurate, high-quality translation in practical domains. In *Proceedings of the 14th conference on Computational linguistics-Volume 3*, pages 1069–1073, 1992.

[10] MD Okpor. Machine translation approaches: issues and challenges. *International Journal of Computer Science Issues (IJCSI)*, 11(5):159, 2014.

[11] Harold Somers. Example-based machine translation. *Machine Translation*, 14 (2):113–157, 1999.

[12] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2): 79–85, 1990.

[13] David Knoke, Peter J Burke, and Peter Burke. *Log-linear models*, volume 20. Sage, 1980.

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

[15] Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. Neural machine translation advised by statistical machine translation. In *AAAI*, pages 3330–3336, 2017.

[16] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*, 2016.

[17] Pasindu Tennage, Prabath Sandaruwan, Malith Thilakarathne, Achini Herath, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. Neural machine translation for sinhala and tamil languages. In *International Conference on Asian Language Processing (IALP), 2017*, pages 189–192, 2017.

[18] Marta R Costa-Jussa and José AR Fonollosa. Latest trends in hybrid machine translation and its applications. *Computer Speech & Language*, 32(1):3–10, 2015.

[19] Nizar Habash, Bonnie Dorr, and Christof Monz. Symbolic-to-statistical hybridization: extending generation-heavy machine translation. *Machine Translation*, 23(1):23–63, 2009.

[20] Catherine Dove, Olga Loskutova, and Ruben de la Fuente. What's your pick: Rbmt, smt or hybrid. In *Proceedings of the tenth conference of the Association for Machine Translation in the Americas (AMTA 2012). San Diego, CA*, 2012.

[21] Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206. Association for Computational Linguistics, 2007.

[22] Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, page 508. Association for Computational Linguistics, 2004.

[23] Lluís Formiga Fanals, Adolfo Hernández Huerta, José Bernardo Mariño Acebal, and Enrique Monte Moreno. Improving english to spanish out-of-domain translations by morphology generalization and generation. In *Proceedings of the Monolingual Machine Translation-2012 Workshop*, pages 6–16, 2012.

[24] Hua Wu and Haifeng Wang. Improving statistical word alignment with a rule-based machine translation system. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

[25] Adam Lopez. Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3):8, 2008.

[26] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54, 2003.

[27] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[28] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[29] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

[30] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[31] Qin Gao and Stephan Vogel. Parallel implementations of word alignment tool. In *Software engineering, testing, and quality assurance for natural language processing*, pages 49–57, 2008.

[32] Richard Zens, Franz Josef Och, and Hermann Ney. Phrase-based statistical machine translation. In *Annual Conference on Artificial Intelligence*, pages 18–32. Springer, 2002.

[33] Miles Osborne. Statistical machine translation. In *Encyclopedia of Machine Learning*, pages 912–915. Springer, 2011.

[34] Frederick Jelinek. Markov source modeling of text generation. In *The Impact of Processing Techniques on Communications*, pages 569–591. Springer, 1985.

[35] Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987.

[36] William A Gale and Geoffrey Sampson. Good-turing frequency estimation without tears. *Journal of quantitative linguistics*, 2(3):217–237, 1995.

[37] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *icassp*, volume 1, page 181e4, 1995.

[38] Frederick Jelinek. Interpolated estimation of markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

[39] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3 (Feb):1137–1155, 2003.

[40] Minwei Feng, Jan-Thorsten Peter, and Hermann Ney. Advancements in reordering models for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 322–332, 2013.

[41] Chi-Ho Li, Dongdong Zhang, Mu Li, Ming Zhou, and Hailei Zhang. An empirical study in source word deletion for phrase-based statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 1–8, 2008.

[42] Kevin Knight. Decoding complexity in word-replacement translation models. *Computational linguistics*, 25(4):607–615, 1999.

[43] Franz Josef Och, Nicola Ueffing, and Hermann Ney. An efficient a* search algorithm for statistical machine translation. In *Proceedings of the workshop on Data-driven methods in machine translation-Volume 14*, pages 1–8, 2001.

[44] 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-Volume 14*, pages 1–8, 2001.

[45] Ye-Yi Wang and Alex Waibel. Decoding algorithm in statistical machine translation. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 366–372, 1997.

[46] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167, 2003.

[47] Abraham Ittycheriah and Salim Roukos. Direct translation model 2. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 57–64, 2007.

[48] Eva Hasler, Barry Haddow, and Philipp Koehn. Margin infused relaxed algorithm for moses. *The Prague Bulletin of Mathematical Linguistics*, 96:69–78, 2011.

[49] Colin Cherry and George Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, 2012.

[50] Mark Hopkins and Jonathan May. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, 2011.

[51] Philipp Koehn and Christof Monz. Manual and automatic evaluation of machine translation between european languages. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 102–121, 2006.

[52] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, 2002.

[53] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002.

[54] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of seventh conference of the association for machine translation in the Americas*, pages 223—231, 2006.

[55] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, 2007.

[56] Ruvan Weerasinghe. A statistical machine translation approach to sinhala-tamil language translation. *Towards an ICT enabled Society*, page 136, 2003.

[57] Sakthithasan Sripirakas, AR Weerasinghe, and Dulip L Herath. Statistical machine translation of systems for sinhala-tamil. In *Advances in ICT for Emerging Regions (ICTer), 2010 International Conference on*, pages 62–68, 2010.

[58] Randil Pushpananda, Ruvan Weerasinghe, and Mahesan Niranjan. Sinhala-tamil machine translation: Towards better translation quality. In *Australasian Language Technology Association Workshop 2014*, volume 129, pages 129–133, 2014.

[59] S Rajpirathap, S Sheeyam, K Umasuthan, and Amalraj Chelvarajah. Real-time direct translation system for sinhala and tamil languages. In *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, pages 1437–1443, 2015.

[60] Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):3, 2007.

[61] Philipp Koehn and Josh Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the second workshop on statistical machine translation*, pages 224–227, 2007.

[62] Nicola Bertoldi and Marcello Federico. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the fourth workshop on statistical machine translation*, pages 182–189, 2009.

[63] Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of EAMT*, volume 2005, pages 133–142, 2005.

[64] Gregor Thurmair. Comparing rule-based and statistical mt output. In *The Workshop Programme*, page 5, 2004.

[65] George Foster and Roland Kuhn. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, 2007.

[66] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[67] Jorge Civera and Alfons Juan. Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 177–180, 2007.

[68] Matthias Eck, Stephan Vogel, and Alex Waibel. Language model adaptation for statistical machine translation based on information retrieval. In *Proceedings of the fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 327—330, 2004.

[69] Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 1(1):3–33, 2002.

[70] Robert C Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224, 2010.

[71] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362, 2011.

[72] Mārcis Pinnis. *Terminology Integration in Statistical Machine Translation*. PhD thesis, Faculty of Computing, University of Latvia, Riga, Latvia, 2015.

[73] Zhixiang Ren, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang. Improving statistical machine translation using domain bilingual multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 47–54, 2009.

[74] Marine Carpuat and Mona Diab. Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 242–245, 2010.

[75] Dhouha Bouamor, Nasredine Semmar, and Pierre Zweigenbaum. Identifying bilingual multi-word expressions for statistical machine translation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 674–679, 2012.

[76] Mahmoud Ghoneim and Mona Diab. Multiword expressions in the context of statistical machine translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1181–1187, 2013.

[77] Raivis Skadiņš, Mārcis Pinnis, Tatiana Gornostay, and Andrejs Vasiļjevs. Application of online terminology services in statistical machine translation. *Proceedings of the XIV Machine Translation Summit*, pages 281–286, 2013.

[78] Mihael Arcan, Marco Turchi, Sara Tonelli, and Paul Buitelaar. Enhancing statistical machine translation with bilingual terminology in a cat environment. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2014)*, pages 54–68, 2014.

[79] Michael Carl and Philippe Langlais. An intelligent terminology database as a pre-processor for statistical machine translation. In *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology-Volume 14*, pages 1–7, 2002.

[80] Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT*, pages 1–8, 2003.

[81] Marcis Pinnis. Dynamic terminology integration methods in statistical machine translation. In *Proceedings of the Eighteenth Annual Conference of the European Association for Machine Translation (EAMT 2015)*, pages 89–96, 2015.

[82] Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh's phrase-based machine translation systems for wmt-14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 97–104, 2014.

[83] Nizar Habash. Four techniques for online handling of out-of-vocabulary words in arabic-english statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 57–60, 2008.

[84] Nizar Habash and Hayden Metsky. Automatic learning of morphological variations for handling out-of-vocabulary terms in urdu-english machine translation. *Proceedings of the Association for Machine Translation in the Americas (AMTA-08)*, pages 107–116, 2008.

[85] Pratyush Banerjee, Sudip Kumar Naskar, Johann Roturier, Andy Way, and Josef Genabith. Translation quality-based supplementary data selection by incremental update of translation models. *Proceedings of COLING 2012*, pages 149–166, 2012.

[86] Chris Callison-Burch, Philipp Koehn, and Miles Osborne. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24, 2006.

[87] Yuval Marton, Chris Callison-Burch, and Philip Resnik. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 381–390, 2009.

[88] Majid Razmara, Maryam Siahbani, Reza Haffari, and Anoop Sarkar. Graph propagation for paraphrasing out-of-vocabulary words in statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1105–1115, 2013.

[89] Chenhui Chu and Sadao Kurohashi. Paraphrasing out-of-vocabulary words with word embeddings and semantic lexicons for low resource statistical machine translation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 644–648, 2016.

[90] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[91] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[92] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics*, pages 86–90, 1998. ISBN 978-88-07-72177-9. doi: 10.3115/980845.980860.

[93] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB : The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, 2013. ISBN 9781937284473.

[94] Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*, 2014.

[95] Hanna Béchara, Yanjun Ma, and Josef van Genabith. Statistical post-editing for a statistical MT system. In *MT Summit XIII*, volume 13, pages 308–315, 2011.

[96] John Hutchins. Example-based machine translation: a review and commentary. *Machine Translation*, 19(3-4):197–211, 2005.

[97] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*, 2017.

[98] Riyafa Abdul Hameed, Nadeeshani Pathirennehelage, Anusha Ihalapathirana, Maryam Ziyad Mohamed, Surangika Ranathunga, Sanath Jayasena, Gihan Dias, and Sandareka Fernando. Automatic creation of a sentence aligned sinhala-tamil parallel corpus. In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pages 124–132, 2016.

[99] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, volume 29, pages 31–43, 2012. ISBN 978-2-9517408-7-7.

[100] LoganathanRamasamy OndrejBojar ZdenekŽabokrtskỳ. Morphological processing for english-tamil statistical machine translation. In *24th International Conference on Computational Linguistics*, pages 113–122, 2012.

[101] S Thenmalar, J Balaji, and TV Geetha. Semi-supervised bootstrapping approach for named entity recognition. *arXiv preprint arXiv:1511.06833*, 2015.

[102] Sanjay K Dwivedi and Pramod P Sukhadeve. Machine translation system in indian perspectives. *Journal of computer science*, 6(10):1111, 2010.

[103] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180, 2007.

[104] Spence Green, Daniel Cer, and Christopher Manning. Phrasal: A toolkit for new directions in statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 114–121, 2014.

[105] Philipp Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Conference of the Association for Machine Translation in the Americas*, pages 115–124, 2004.

[106] George Foster, Roland Kuhn, and Howard Johnson. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61. Association for Computational Linguistics, 2006.

[107] Andreas Stolcke. Srilm-an extensible language modeling toolkit. In *Proceedings of Seventh international conference on spoken language processing*, pages 901–904, 2002.

[108] Liang Huang and David Chiang. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 144–151, 2007.

[109] Dietrich Klakow. Log-linear interpolation of language models. In *Fifth International Conference on Spoken Language Processing*, 1998.

[110] Anthony Rousseau. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100: 73–82, 2013.

[111] Arianna Bisazza, Nick Ruiz, and Marcello Federico. Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. *International Workshop on Spoken Language Translation (IWSLT)*, pages 136—143, 2011.

[112] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[113] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[114] Mokanarangan Thayaparan, Surangika Ranathunga, and Uthayasanker Thayasivam. Graph Based Semi-Supervised Learning Approach for Tamil POS tagging. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

[115] Utpal Sharma, Jugal K Kalita, and Rajib K Das. Acquisition of morphology of an indic language from text corpus. *ACM Transactions on Asian Language Information Processing (TALIP)*, 7(3):9, 2008.

[116] Philipp Koehn and Hieu Hoang. Factored translation models. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[117] Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis Leiva, et al. Casmacat: A computer-assisted translation workbench. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28, 2014.

[118] Prashant Mathur, Cettolo Mauro, and Marcello Federico. Online learning approaches in computer assisted translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 301–308, 2013.