Manipuri-English Machine Translation using Comparable Corpus

(An Unsupervised Statistical Machine Translation Approach)

Thesis submitted in partial fulfilment of the requirements for the award of the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Lenin Laitonjam

Under the supervision of

Dr. Sanasam Ranbir Singh

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati - 781039 Assam India

June, 2022

Dedicated to
My beloved Parents
and
My dear Brother

Acknowledgments

First and foremost, I would like to convey my heartfelt appreciation to my Ph.D. supervisor Dr. Sanasam Ranbir Singh, for his unwavering support and guidance. His steadfast support and mentoring have been crucial in my development as a scholar and person. I am glad for the opportunity to work with him and will be eternally grateful to him.

I would also like to thank Dr. Vijaya Saradhi, Prof. Laishram Boeing Singh, and Prof. Sukumar Nandi from my doctorate committee for their insightful remarks, which improved the quality and clarity of my work. I want to thank all the Department of CSE faculty members for their direct and indirect support. I also sincerely thank the technical and administrative team of the department for their ongoing assistance and support.

I would also want to thank all of my friends and seniors at IITG. Gyanendro, Neelakshi, Rajlakshmi, Ranjan, Hemanta, Nanaobi, Bidyalashmi, Bornali, Durgesh, Akash, Deepen, Jennil, and many more in the OSINT lab deserve special thanks and gratitude. They have been extremely supporting and encouraging. I would also like to specifically mention Monica, Biki, Anil, Menan, and Debeni for helping me annotate the data. Without them this work would have not been possible to complete. I am also thankful to all my colleagues at NIT Mizoram for their courteous help and support throughout my study period.

I am blessed to have good buddies - Somorjit, Shufen, Bale, Birjit, Gishan, Alex, Rakesh, Burnish, and Kishan, with whom I have shared some indelible moments of my life at IITG. I have enjoyed gathering with the IITG Manipuri family for all the events we have organized and participated in. They have made my stay at IITG lively and fun-filled.

I want to express my gratitude to Anil, Amarjit, Henry, James, Sanatomba, Nirosh, Ashin, Baleshwor, Bidanta, Birlakh, and Malemnganba, my undergraduate and school days pals, for the wonderful memories and cherished times. Last but not least, my heartfelt appreciation to my family - my parents and my brother - for being with me in every step. Your prayer made me sustained this far.

Lenin Laitonjam
166101018

Declaration

I certify that

- The work contained in this thesis is original and has been done by myself and under the general supervision of my supervisor.
- The work reported herein has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.
- I also affirm that no part of this thesis can be considered plagiarism to the best of my knowledge and understanding and take complete responsibility if any complaint arises.
- I am fully aware that my thesis supervisor are not in a position to check for any possible instance of plagiarism within this submitted work.

Lenin Laitonjam
166101018

Certificate

This is to certify that this thesis entitled "Manipuri-English Machine Translation using Comparable Corpus (An Unsupervised Statistical Machine Translation Approach)" submitted by **Lenin Laitonjam** (166101018), in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy, to the Indian Institute of Technology Guwahati, Assam, India, is a record of the bonafide research work carried out by him under my guidance and supervision at the *Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam, India.* To the best of my knowledge, no part of the work reported in this thesis has been presented for the award of any degree at any other institution.

Dr. Sanasam Ranbir Singh
Associate Professor
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati-781039, Assam, India

Manipuri-English Machine Translation using Comparable Corpus

(An Unsupervised Statistical Machine Translation Approach)

Abstract

Machine Translation (MT) is an essential tool for communicating with foreign-language speakers. The current mainstream MT frameworks, namely, Statistical MT (SMT) and Neural MT (NMT), are characterized by learning to translate automatically via machine learning techniques. It has been observed that these systems require a large number of parallel sentences between the source and the target language pair to produce a high-quality translation. Unfortunately, readily available parallel sentences are limited for most language pairs. Manually generating a quality parallel corpus is also very costly and time-consuming. As a result, many practical applications of MT are restricted to widely spoken and rich-resource languages. On the other hand, MT quality has not reached a reasonable level in many low-resource language pairs.

This thesis reports the problem of developing an MT system that translates between low-resource Manipuri and English. Manipuri is one of the scheduled Indian languages. The study focus on improving the MT quality between the language pair by exploiting unsupervised MT approaches to cope with bilingual corpora's scarceness. Unsupervised MT enables translation between languages without using parallel data by exploiting source and target language monolingual corpora. Although various unsupervised methods have been proposed, studies have shown their quality decreases with the difference in the domain between the source and the target languages corpora. This thesis first presents a Manipuri-English comparable corpus to facilitate MT research between the language pair. The corpus belongs to the same domain and is also aligned at date and document levels. Preliminary investigation results show that the proposed corpus is feasible for developing MT systems for the language pair. It is also observed that out of the two main unsupervised MT approaches; standard unsupervised SMT model performs superior than unsupervised NMT models on the language pair.

Although the results are promising, unsupervised MT techniques have the drawback that their performance suffers when the source and target languages have different linguistic properties. To alleviate issues incurred due to different linguistic aspects between English and Manipuri, this thesis proposes two methods.

The first method is proposed to normalize the morphological inflection issue of Manipuri. The study aims to deploy a Manipuri suffix segmenter for the problem. Unfortunately, there is no publicly available suffix segmenter/morphological analyzer for the language. This thesis also presents a Manipuri Suffix Segmenter to segment inflected Manipuri words into root and suffixes. From various experimental results, it is observed that segmenting the text corpus significant improves the performance. The second method is proposed to induce inter-language connecting points between Manipuri and English. This thesis developed a transliteration model to produce transliteration features that will enable the unsupervised MT models to exploit the phonetically similar vocabularies between the language pair. Experimental results show that incorporating transliteration features improves translation results over the corresponding baselines. The study also proposes a hybrid machine transliteration model to transliterate English loanwords and named-entities to Manipuri. The proposed hybrid model improves traditional encoder-decoder transliteration methods by incorporating a multi-source framework that leverages grapheme and phoneme sequences.

The last part of the thesis is dedicated to making the best use of the proposed comparable corpus for the language pair MT task. Specifically, the study exploited the document-aligned and temporally-aligned characteristics of the corpus. Firstly, this thesis proposes a multi-step approach to exploit document-aligned comparable corpus. First, similarity scores between source and target phrases based on the document-aligned characteristics of the comparable corpus are obtained. Then, the similarity scores are incorporated into the unsupervised SMT model. Secondly, a novel method to generate temporal cross-lingual embedding is proposed to exploit the temporal-aligned corpus. The proposed embeddings assist in developing more robust source and target phrases alignments and increase the overall translation performance. From various experimental results on English-to-Manipuri and Manipuri-to-English MT, it is observed that both the proposed methods developed for leveraging the comparable corpus's different alignment characteristics succeeded in their respective task and further enhanced the translation results.

Contents

Listing of figures

List of Tables

1

Introduction

Machine translation (MT) is a natural language processing task of translating texts or speeches in one language to another language automatically by using computer programs. MT has become vital in today's contemporary, increasingly globalized society [48]. Widespread use of the internet and rapid growth of web materials have also prompted increasing demands for automated machine translation systems. With automated MT systems, internet users will easily comprehend content from different languages leading to a more effective way of sharing

knowledge without language barriers. Apart from personal use, MT will also assist business houses in reaching out to global audiences and expanding to global markets.

This thesis focuses on developing an automated MT system for Manipuri-English language pair. Manipuri is a Tibeto-Burman language spoken primarily in the Indian state of Manipur, and small populations in the neighboring states (Assam and Tripura) and neighboring countries (Myanmar and Bangladesh). Manipuri is used as a lingua-franca language in the state of Manipur and is listed as one of the VIII scheduled languages in the Indian constitution. It is locally known as *Meiteilon*, derived from the term *Meitei* (majority community in Manipur) and *Lon* (Language). Though Manipuri is one of the oldest languages in South East Asia*, it is still considered to be a low-resource language in terms of digitized text resources and language processing tools, automated MT in particular, as compared to other major languages of India. An advancement in developing automatic MT for Manipuri-English language pair will boost the Manipuri community in various aspects allowing them to communicate with the rest of the world. The impact of automatic MT on Manipur's tourism industry is also massive, as it demands an easy and effective communication channel between tourists and locals. Therefore, developing an effective automated MT system becomes an important task from various spectrums of applications. Although MT is applicable for both the text and speech, the scope of this thesis is limited to the textual domain. Manipuri is written using two scripts; *Bengali Script*† and its native *Meitei Mayek*‡. Consider-

*http://gmj.manipal.edu/issues/november2018/effect-of-shifting-orthographic-\practices-of-manipuri-script-on-millennials.pdf

†https://en.wikipedia.org/wiki/Bengali_alphabet

‡https://en.wikipedia.org/wiki/Meitei_script

ing that most of the written documents are available in Bengali script, this thesis considers Manipuri texts written in Bengali script.

Developing an effective MT system is not a trivial task; needing various resources such as a large parallel corpus, parser, stemmer, morphological analyzer, bilingual dictionary, spell-checker, etc. Initial studies on MT consider rule-based models [157] by using various language-dependent tools and resources like syntactic parser [9, 36], inter-lingual representation [53], bilingual dictionary [91], etc. Considering the limitations of using language-dependent rules, which further depend on other language-dependent tools and resources, researchers have come up with data-driven approaches like Statistical Machine Translation (SMT) [115] and Neural Machine Translation (NMT) [16]. Since the invention of SMT and NMT methods, the attention of researchers from the machine learning domain has been drawn, and several machine learning-based SMT and NMT methods have been proposed in the literature. However, one of the challenging concerns for developing MT systems for low-resource languages is creating a large volume of parallel sentences, which is a manual and expensive task requiring professional translators fluent in both the source and target languages. Even for rich-resourced languages, the concern is still valid for domain-specific MT systems such as medicine, scientific papers, etc.

To overcome the issue of creating a large volume of a sentence-level parallel corpus, researchers have recently proposed unsupervised SMT and NMT methods without the need for a large sentence-level corpus. Studies [14, 43] have shown that such unsupervised approaches can also be effectively used for developing automatic MTs. Motivated by such observations, this thesis focuses on developing an automatic Manipuri-English MT system using unsupervised approaches over

a *comparable corpus*. A comparable corpus is a collection of two monolingual corpora which share common characteristics such as topic, subject, domain, thematic, genre, sampling period, and so on but are not exact translations of each other. Considering the need for large monolingual corpora and comparatively lower performance while using unsupervised NMT compared to its unsupervised SMT counterpart (as observed in our preliminary investigation reported in Section 6.3), the scope of this thesis is limited to unsupervised SMT.

1.1 Manipuri MT: Its status and challenges

Manipuri is still considered a low-resource language. The availability of digitized text resources and language processing tools for various NLP applications are still in their nascent stages. Only a few studies on the development of Manipuri MT can be found in the literature. Table 1.1 shows a list of studies reported in the literature for developing different NLP tools, including MT. Majority of these studies consider either a dictionary or rule-based approach due to data constraints. Bilingual resources of the Manipuri language are close to non-existent. Although several efforts have been made to compile parallel sentences between different Indian languages, there is limited language coverage [39, 209, 94, 75, 220]. To the best of our knowledge, the corpus presented in the studies [94, 75] are the only publicly available parallel sentences for the Manipuri-English language pair. TDIL-corpus* [94] consists of around 11k sentences in the tourism domain, while only 7484 sentence pairs were recently generated for the language pair from the website of the Prime Minister of India† in [75]. Few researchers have reported the

*https://www.tdil-dc.in
†www.pmindia.gov.in

Table 1.1: Studies reported for developing different NLP tools, including MT for Manipuri language.

Paper	Year	Tool	Methodology
[218]	2012	POS Tagger	Rule-based
[158]	2018	Parser	Rule-based
[40]	2004	Morphological Analyzer	Rule and Dictionary based
[213]	2008	POS Tagger	Dictionary-based
[212]	2006	Word Class and Sentence Type Identification	Dictionary-based
[161]	2011	Stemmer	Dictionary-based
[216]	2008	POS Tagger	CRF (Conditional Random Field) and SVM (Support Vector Machine)
[217]	2009	Named Entity Recognition (NER)	SVM
[175]	2013	Keywords Spotting	Hidden Markov Model (HMM)
[189]	2016	Pronunciation Dictionary	CRF, HMM, and Maximum-entropy Markov model (MEMM)
[204]	2016	Syllabification	Hybrid-based
Manipuri-English MT			
Paper	**Year**	**Methodology**	**Description**
[214]	2010	Example-based MT	Morphological analysis, NER, POS tagging, and chunking are also applied on the examples.
[203]	2010	Factored SMT	Incorporate rule-based morpho-syntactic and semantic information.
[215]	2011	Phrase-based SMT	Integrate reduplicated multi-word expressions and named-entities.
[210]	2013	Phrase-based SMT	Investigate which Manipuri Script (Bengali or Meitei Mayek) is better for English-Manipuri Pair.

development of Manipuri-English MT systems [214, 203, 215, 210], as presented in Table 1.1. For example, authors in [214] conducted a thorough investigation of the effects of morpho-syntactic information and dependency relationships on an SMT model. Singh & Bandyopadhyay [215] demonstrated that integrating linguistic variables such as named-entities and reduplicated multi-word expressions improves the phrase-based SMT system. However, these studies used in-house generated datasets, and they are not publicly available.

Apart from resource constraints, language-specific challenges need to be ad-

dressed while building an MT system for the Manipuri and English language pair. In the following subsection, some of the critical linguistic characteristics that may affect Manipuri MT development are discussed.

1.2 Few Linguistic Characteristics Related to MT Development

1. **Morphological Richness**: Manipuri, like most other Indian languages, is highly agglutinative, having very rich morphological structures [29]. The language tends to generate lots of new words derived from a single root word. Words are primarily associated with suffixes depending on the number, gender, etc. [161, 204]. Suffixes are more prominent than the prefixes, while there are no infixes. There are words with as many as 10 (ten) suffixes attached to a single root [161]. Such inflections result in many unseen and low-frequency words.

2. **Word Order:** As opposed to English's Subject-Verb-Object word order, Manipuri generally follows the Subject-Object-Verb word order [200, 159]. The following examples show the difference in word order between English sentences and the corresponding Manipuri translation.

    - Tomba goes to school

      তোম্বা (tomba) মহৈরোইশংদা (to school) চতলি (goes)

      tomba maheiloisangda chatli[*]

[*]The italic word/phrase below each Manipuri word/phrase represents their transliteration in the Roman alphabet.

- They went to the market

  মাখোয় (they) কৈথেল্দা (to the market) চতলুই (went)

  makhoi keithelda chatlui

Some of the other prominent features of Manipuri word order which may also affect development of MT are listed below with examples.

(a) Conditional clause generally precedes the main clause.

- If you complete the work, I will give you something.

  নঙ থবক অসি লোইশল্লবদি (If you complete the work) , ঐ নঙোন্দা কৈনোম পিগে (I will give you something) ।

  nang thabak asi loisallabadi, ei nangaonda keinom pige

(b) Adverb generally precedes verb.

- Sana will come quickly.

  সানা (Sana) থুনা (quickly) লাক্কনি (will come) ।

  sana thuna lakkani

(c) Main clause are generally preceded by the subordinate clause.

- They waited for him while he was eating.

  মানা চাকচরি ঙেদা (while he was eating), মখোইনা মাবু ঙাইরম্বনি (they waited for him) ।

  maana chakchari ngeida, makhoina mabu ngairambani

(d) Descriptive adjective can either follows or precedes the noun.

- big dog

  অচৌবা (big) হুই (dog)

  achouba hui

ꯍꯩꯃꯤ (dog) ꯑꯆꯣꯕꯥ (big)

hui achouba

(e) Demonstrative and the numerical adjectives generally follow the noun.

- this dog

  ꯍꯩꯃꯤ (dog) ꯑꯁꯤ (this)

  hui asi

- two woman

  ꯅꯨꯄꯤ (woman) ꯑꯅꯤ (two)

  nupi ani

(f) For the combination of noun and adjective, the adjective always form the final constituent of the compound word.

- big dog

  ꯍꯩꯖꯥꯎ

  huijao

(g) In the noun phrase, when all the three modifiers, i.e., demonstrative, descriptive, and numerical adjective, are present, there are two possible combinations.

i. When the adjective follows the noun, the order is *noun-descriptive-numeral-demonstrative.*

- these two tall man

  ꯅꯨꯄꯥ (man) ꯑꯔꯥꯡꯕꯥ (tall) ꯑꯅꯤ (two) ꯑꯁꯤ (these)

  nupi awangba ani asi

ii. When the adjective precedes the noun, the order is *descriptive-noun-numeral-demonstrative*.

- these two tall man

  অৱাঙবা (tall) নুপা (man) অনি (two) অসি (these)

  awangba nupi ani asi

(h) Manipuri generally has post-positions rather than prepositions.

- under the table

  তেবলগী (the table) মখাদা (under)

  tablegi makhada

1.3 Approaches towards Low-resource Machine Translation

Over the years, researchers have devised several strategies to overcome the data scarcity problem in MT development for low-resource environments. Some of the notable approaches include data augmentation [70, 93] and multi-lingual MT [253, 98, 69, 46]. However, these approaches still rely on hundreds of thousands of parallel sentences [197], large document-aligned comparable corpora [179], bilingual dictionaries [232], etc. Since adequate amounts of parallel data for low-resource languages like Manipuri are still a big concern, it is crucial to find technological solutions that compensate for these shortcomings. Motivated by this, unsupervised MT (UMT) models, namely Unsupervised Statistical Machine Translation (USMT) [127, 12] and Unsupervised Neural Machine Translation (UNMT) [224, 42] are developed recently. They are a special class of MT model that depends only on source and target language monolingual corpora. Such systems generally exploit unsupervised cross-lingual embeddings [43, 11]. These

unsupervised models have achieved remarkable results, even outperforming the supervised attention-based NMT model [16] for English-French translation [224].

Despite reported successes, UMT is still in its nascent stage, and its ability to handle low-resource environments is still an open research problem. Majorities of the successfully developed previous unsupervised MT-related studies [127, 14, 42, 224, 13, 182, 183] are performed for rich resource languages like English, German, French, etc., where high-quality monolingual corpora are also available in abundance. Studies in [143, 108] have already shown that the effectiveness of such methods decreases with the increase in domain variation between the source and target monolingual corpora. Studies in [143, 131] have also reported that unsupervised models do not perform well for the languages with different language characteristics such as language branch, alphabet, morphology, etc. Marchisio et al. [143], for example, found that the gap in performance between supervised and unsupervised methods for the two distant languages*, Russian and English pair, is larger than that of the comparable French and English language pair. Similar observations are also made for other distant and low resource language pairs, namely, Sinhala-English and Nepali-English. Motivated by such observations, this thesis investigates the effectiveness of UMT models for the Manipuri-English language pair, another distant and low-resource language pair with limited monolingual data, and proposes potential solutions using a document and temporal level aligned *comparable corpora*.

*Languages with contrasting linguistic characteristics like language branch, alphabet, morphology, etc.

1.4 Research Objectives

The primary goal of this thesis is to *develop an effective MT system for the distant Manipuri-English language pair using comparable corpora in lieu of expensive parallel sentences.*, with the following objectives.

1. *To generate a cost-effective bilingual corpus to aid MT studies between the Manipuri and English languages.* This thesis creates a comparable news corpus curated from publicly available Manipuri news sources.

2. *Investigate the performance of different state-of-the-art UMT models (both statistical and neural) on Manipuri-English comparable corpus, and identify underlying challenges.* Empirical observations show that USMT model significantly outperforms its UNMT counterpart on our experimental comparable corpus. Therefore, the thesis focuses on extending USMT framework.

3. An effective cross-lingual embedding is one of the core problems in UMT. Further, most of the dominating cross-lingual embedding relies on a bilingual dictionary. Assuming that a Manipuri-English bilingual dictionary may not be readily available, the thesis proposes a novel idea of using *transliterated word pairs* instead of a bilingual dictionary.

4. *Enhance the performance of USMT by exploiting different level of alignments (documents and temporal) on the comparable corpus and usage of transliteration word pairs.*

1.5 Contributions

This thesis makes the following contributions in line with the above research objectives.

1.5.1 Dataset Generation

As stated above, the Manipuri-English language pair lacks the requisite corpus for MT research. As a result, this study first creates two essential datasets.

1. **Manipuri-English Comparable Corpus**

   A new *Manipuri-English comparable* corpus is created by curating texts from two publicly available news sources on the internet, namely *Sangai Express* and *Poknapham*, two local news publications in Manipur. This dataset is the first of its kind for Manipuri-English language pair. The corpus is aligned at the document as well as temporal level.

2. **Manipuri-English MT Evaluation Dataset**

   This thesis also creates a small sentence-level parallel corpus for evaluating the performance of the MT systems reported in this thesis.

1.5.2 Building Dependent Tools

Some of the necessary language processing tools for developing Manipuri MT are not publicly available. This thesis further develops the following tools.

1. **Manipuri Suffix Segmenter**

   The thesis proposes an effective Manipuri suffix segmenter to normalize

the agglutinative nature of Manipuri text by adapting a popular language-independent Stemming algorithm, namely GRAS (GRAph-based Stemmer) [166].

2. **Transliteration Models**

   The thesis proposes a multi-sources encoder-decoder based neural network model to machine transliterate English loanwords and named-entities to Manipuri by combining the advantages of both grapheme and phoneme representations of the texts simultaneously.

1.5.3 Manipuri SMT using Comparable Corpus

A preliminary evaluation of existing UMT models shows that the state-of-the-art USMT model, namely Monoses [12], significantly outperforms its UNMT counterpart on our experimental comparable corpus. As a result, the thesis considers Monoses and modifies it to further enhance the translation performance for Manipuri-English language pair. It makes the following three major contributions.

1. **Incorporating Transliteration Features**

   Exploit transliteration word pairs, instead of bilingual dictionary, to improve

   - the cross-lingual embedding between Manipuri-English languages.
   - the phrase-table mapping required for Manipuri-English MT.

2. **Document level alignment of the Comparable Corpus**

   As document-aligned pairs describe a common news event, the chances of obtaining translated word/phrase pairs between the source and target documents are higher. The thesis exploits this assumption to enhance the translation performance.

3. **Temporal alignment of the Comparable Corpus**

   Though the probability of obtaining translated word pairs from the above document-aligned corpus is intuitively higher, it does not enhance the performance significantly due to data sparsity. To overcome this problem, the thesis extends the alignment to wider temporal windows (like weekly, monthly, etc.) and proposes a novel method to enhance the translation performance.

1.6 Organization of the Thesis

- **Chapter 2: Literature Reviews and Background Studies**

  This chapter provides an overview of the SMT and NMT approaches, followed by a survey on various approaches proposed for adapting SMT and NMT for low-resource scenarios. Finally, we offer detailed background studies on the fundamentals and current trends in unsupervised MT.

- **Chapter 3: Manipuri-English Comparable Corpus**

  This chapter describes the processes for building a comparable corpus feasible for cross-lingual studies between Manipuri and English language pairs.

- **Chapter 4: Transliteration of English Loanwords and Named-entities to Manipuri**

  In this chapter, we describe our contributions related to the development of the transliteration model.

- **Chapter 5: Empirical Study of Unsupervised Cross-lingual Embedding Methods**

  This chapter provides an preliminary evaluation of two popular unsupervised

approaches of inducing cross-lingual word embeddings, namely MUSE [43] and Vecmap [11], on the language pair. The proposed Manipuri Suffix Segmenter for normalizing the morphological inflections issue and the method of enhancing cross-lingual embeddings using transliterated word pairs are also discussed in this chapter.

- **Chapter 6: Manipuri-English MT using a Comparable Corpus**

  This chapter provides a detailed description of the method developed for enhancing the USMT model by incorporating (i) suffix segmenter, (ii) transliteration features, and (iii) a method proposed for exploiting document-aligned comparable corpus.

- **Chapter 7: Improving Manipuri-English MT by Exploiting a Temporally Aligned Comparable Corpus**

  This chapter discusses the proposed method that exploits the temporal-aligned characteristics of the comparable corpus for improving the MT performance.

- **Chapter 8: Conclusions and Future Works**

  This chapter presents our concluding remarks on the thesis work and some of the potential directions to work in the future.

2

Background Studies and Literature Reviews

This chapter provides a review of existing MT approaches. We start by giving background studies related to statistical MT (SMT) and neural MT (NMT), which serve as the foundation for unsupervised MT (UMT) models. The following subsections give an overview of various major strategies proposed for adapting SMT and NMT for low-resource scenarios. The last section of this chapter goes

through the basics and current trends in UMT approaches.

2.1 Background

Despite a long history of research, MT is yet to achieve its initial goal of replacing human translators. MT demands a discourse understanding and interpretation of the sentences. It requires capturing the speaker's intentions and mental status. In some cases, MT may also require common sense and world knowledge for translation. It involves both natural language understanding and generation framework. Consequently, developing a robust translation system demands several expensive resources making it highly challenging for low-resource languages.

Over the years, several approaches have been proposed to produce quality translations. Earlier approaches relied on hand-crafted rules. The advantage of *Rule-Based MT (RBMT)* [91, 53, 9, 36] is that the translation outcomes are not dependent on parallel sentences. On the downside, these approaches require a large number of linguistically motivated rules. Knowledge experts familiar with both the source and target languages are necessary for each language pair to devise the rules. Furthermore, constructing pre-defined rules that account for all of the syntactic and semantic inconsistencies required for a good translation is not a trivial task. *Example-based MT (EBMT)* models [190] is able to reduce the over-reliance on costly hand-written rules. In this method, a source sentence is translated to a target sentence by imitating the translation of similar examples already present in a database. However, EBMT generally relies on word-co-occurrence information, including linguistically motivated annotated data such as part-of-speech, bilingual dictionaries, thesauri, and so on. Moreover, the lack of statistical models to score

the selected examples make the method less versatile.

In recent decades, fully data-driven methodologies have significantly advanced the field of MT. SMT [116] and NMT [38, 16] have become the dominant data-driven MT framework in both theory and practice. Such systems are cost-effective as they can learn translation features using only parallel sentences, thereby considerably reducing the amount of time and resources spent constructing linguistic rules. It has eliminated the overreliance on human experts for modeling the translation process. A detailed description of standard SMT and NMT models is presented below.

2.1.1 Statistical Machine Translation

SMT generates translations based on the combination of several statistical models [115], whose parameters are learnt from a large sentence-aligned translated corpus. In this framework, the problem of generating a target sentence (t) given a source sentence (s) is projected as the problem of searching for the most probable target sentence t that maximizes the conditional probability $p(t|s)$ given by:

$$\hat{t} = \arg\max_{t} p(t|s) \tag{2.1}$$

Using the Bayes' rule, $p(t|s)$ is decomposed as:

$$p(t|s) = \frac{p(t)p(s|t)}{p(s)} \tag{2.2}$$

Since $p(s)$ is independent of t, estimating the best target sentence $\hat{t}$ is same as maximizing the equation:

$$\hat{t} = \arg\max_t p(t|s) = \arg\max_t p(t)p(s|t) \tag{2.3}$$

The equation 2.3 is also called the *Fundamental Equation of Machine Translation*. It consist of the following two major probability models:

1. Language model ($p(t)$):

   This model is responsible of the fluency of the translation output. It assigns probabilities to a sequence of words $w_1, w_2, ...w_n$. The joint probability of the sequence is computed using the chain rule as follows:

$$P(w_1, w_2, ...w_n) = \prod_{i=1}^{n} P(w_i|w_1, ...w_{i-1}) \tag{2.4}$$

2. Translation model ($p(s|t)$):

   Translation model assigns a conditional probability $p(s|t)$ to each source (s) and target (t) sentence pairings. Researchers have considered several ways of encoding s and t to generate robust translations. Phrase-based Statistical MT (PBSMT)[116] decodes the target sentence t as a sequence of n phrases $t_1, t_2, ..., t_n$ corresponding to the source sentence phrases sequence $s_1, s_2, ..., s_n$. The translation model is subsequently decomposed as:

$$p(s|t) = \prod_{i=1}^{n} \varphi(s_i|t_i)d_i \tag{2.5}$$

   where $\varphi(s_i|t_i)$ is the phrase translation model and d_i represents a distor-

tion function or reordering model. This approach facilitates many-to-many alignments enabling the model to capture phrasal cohesiveness naturally. In PBSMT, phrases are not only limited to linguistically motivated phrases like noun phrases, verb phrases, and so on, but non-linguistic phrase pairs are also possible. Hierarchical phrase-based models [35, 239] extend the notion of phrase mapping by defining translation rules as a synchronous context-free grammar. This inclusion of grammar formalism further allows the extension by incorporating linguistic annotations, thereby combining syntactic information to translation rules in syntax-based models [172, 87].

Apart from the translation and language models, a standard SMT system may consist of several other feature models. A log-linear model is usually trained using the Minimum Error Rate Training (MERT) algorithm [163]. The log-linear model assigns a weight λ_i to each feature $f_i(s,t)$, where s and t is the source and target sentences, respectively. Some standard features include language models, forward and backward-translation models, word and phrase penalty scores, etc [115]. Typically, each of these feature models is optimised separately. The target sentence is constructed left-to-right during decoding. Beam search is generally use to find the best translation t that maximises the log-linear model score $\sum_i \lambda_i f_i(s,t)$.

2.1.2 Neural Machine Translation

In recent years, neural network models are becoming extremely popular in NLP studies. NMT is an MT system based on neural network architecture. It is relatively new as compared to its statistical counterpart. However, NMT has outperformed well-established SMT for the majority of the cases if quality par-

allel sentences are available in abundance. Its success can be mainly attributed to distributed language representations, enabling end-to-end training of an MT system. Unlike classical SMT, tuning components such as language model, translation model, distortion model, etc. separately are not required. Among several NMT architectures, Recurrent Neural Network (RNN) and Transformer are two of the most popular NMT models.

I. RNN-based Models

RNN-based NMT models [100, 38, 229, 37] generally follow an encoder-decoder setup. Although the RNN encoder-decoder architecture was proposed to re-score the phrase pairs of PBMT [38], they triggered a positive direction toward the use of neural network technologies for the MT problem. The model has become a standard NMT architecture. The task of an encoder is to understand the input source sentence $x_1, x_2, x_3, ..., x_n$ and generate an encoded representation $\boldsymbol{h}$ of the entire sequence. Suppose, at each time step t, the encoder RNN hidden state is updated as:

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, x_t) \tag{2.6}$$

where f is a non linear function (Long Short Term Memory (LSTM)[88] or Gated Recurrent Unit (GRU)[38]) and $\boldsymbol{h}_{t-1}$ is the previous hidden state. Then, the encoded representation $\boldsymbol{h}$ is the hidden state of the RNN after reading the entire input sequence. The encoded representation is then forwarded to the decoder RNN for generating the target sentence $y_1, y_2, y_3, ..., y_m$. The target sentence is obtained sequentially by predicting each word in the sequence y_t at each time t,

using the current decoder hidden state $\boldsymbol{h}_t^d$ as:

$$p(y_t|y_1, y_2..., y_{t-1}, \boldsymbol{h}) = softmax(g(\boldsymbol{h}_t^d)) \tag{2.7}$$

where g is a transformation function that generates a vector of size equal to the number of graphemes in target language. The decoder hidden state ($\boldsymbol{h}_t^d$) at time t is updated using the previous hidden state $\boldsymbol{h}_{t-1}^d$, encoded representation $\boldsymbol{h}$ and previous predicted output y_{t-1} as:

$$\boldsymbol{h}_t^d = f(\boldsymbol{h}_{t-1}^d, y_{t-1}, \boldsymbol{h}) \tag{2.8}$$

If T is the training sentences consisting of a list of source sentences s and corresponding target sentences t, then the training objective of the model is defined by minimizing the following entropy-loss:

$$J = \sum_{(s,t)\in T} -\log p(t|s) \tag{2.9}$$

Over the years, researchers have considered several variants of encoder-decoder architectural setups for NMT. Schuster & Paliwal [193] have exploited both unidirectional and bidirectional RNN as encoders. Bidirectional RNN is used to enable the encoder to encode both preceding and following contexts for each word on the source sentence [16, 198, 26]. Several authors have also considered stacking multiple layers on top of one another for both the encoder and decoder sides to enhance the translation performances [246, 26, 141]. The motivation behind adopting such deep layer settings is that the model would converse with a better result than a shallow one.

A critical problem with early NMT models is that their performance decreases with an increase in sentence length. Cho et al. [37] suggested that this weakness is due to the fixed-length source sentence encoding. To tackle this problem, Bahdanau et al. [16] introduced the concept of *attention mechanism* to avoid having a fixed-length source sentence representation. NMT uses attention to determine which parts of the input sequence are important to each word in the output, allowing the model to select the optimal output based on relevant information. The attention mechanism has significantly improved the translation performance. Having seen the effect of the attention mechanism, Vaswani et al. [238] proposed an NMT model called the Transformer. At a high level, the model is the same as previous encoder-decoder models. However, transformer use the attention mechanism within the encoder itself called the *self-attention*. On top of the conventional attention mechanism, self-attention aids the encoder encode the sequence much more effectively. A detailed description of the transformer is discussed in the next section.

II. Transformer

The Transformer model [238, 135] takes advantage of the positional embedding to encode word order in a word sequence like a sentence without using any recurrent layer. Here, the encoder consists of a stack of multiple identical layers. Each layer consists of a multi-head attention sub-layer followed by a position-wise feed-forward neural network. Further, both the multi-head and the feed-forward sub-layers are coupled with a normalization layer and a residual connection. The multi-head attention sub-layer computes self-attention weights for each token within a sequence, including the token itself. Self-attention relates different positions of

the input sequence and is computed as follows:

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \tag{2.10}$$

where $\boldsymbol{V}$ is a matrix that contains all the value vectors in the sequence each of dimension d_v. Similarly, $\boldsymbol{Q}$ and $\boldsymbol{K}$ are the queries and keys matrices where each key and query vectors are of dimension d_k.

In practice, the multi-head attention calculates self-attention h times, where h is the head number. That is, the vectors that represent the queries, keys, and values are linearly transformed to h number of projections. The attention in each head is computed independently, and then the outputs are concatenated and projected back to the original dimension as:

$$Multihead(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = (\boldsymbol{head}_1 : ... : \boldsymbol{head}_h)\boldsymbol{W}^o \tag{2.11}$$

$$\boldsymbol{head}_i = Attention(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V) \tag{2.12}$$

where d_{model} is the model dimension, and $\boldsymbol{W}_i^Q \in R^{d_{model} \times d_k}$, $\boldsymbol{W}_i^K \in R^{d_{model} \times d_k}$, $\boldsymbol{W}_i^V \in R^{d_{model} \times d_v}$ and $\boldsymbol{W}^o \in R^{hd_v \times d_{model}}$ are parameter matrices. Each encoder multi-head attention layer takes outputs of the previous layer as inputs to enable it to attend to all the positions of the previous layer. The decoder has a similar architectural structure to the encoder, stacking multiple identical layers. Each layer consists of multi-head attention followed by a feed-forward network coupled with layer normalization and residual connection. However, there are two multi-head attention sub-layers in the decoder: i) a decoder masked multi-head attention and

ii) encoder-decoder attention. The decoder masked multi-head attention layer attends to the previous predictions and masks the future positions. In the second decoder multi-head attention sub-layer (encoder-decoder attention), keys and values come from the output of the encoder. In contrast, the queries come from the previous decoder layer. Thus, enabling every position in the decoder to attend overall positions in the input sequence.

2.2 Low-resource Machine Translation

Although data-driven MT discussed above has achieved near human-level performance in some languages, they rely on a large amount of parallel sentences [46, 238]. As a result, progress in this field is generally confined to resource-rich languages. To address this issue, several strategies have been proposed for low-resource MT. In the subsequent section, we discuss some of the most prominent techniques, namely, *Data Augmentation* and *Multi-lingual MT*.

2.2.1 Data Augmentation

Data augmentation techniques create additional data by modifying existing data or adding data from other in-expensive sources to enhance the original parallel dataset. The techniques are applicable to both the SMT and NMT models. Researchers have generally considered utilizing monolingual data to improve the quality of MT systems trained with a modest number of parallel sentences [70, 197]. Earlier works considered integrating language model (LM) trained on monolingual data to improve the fluency of the generated text [70, 93]. On the contrary, Sennrich et al. [197] proposed *Back-translation* in which they populate the parallel

sentences by adding target-side monolingual data, and its corresponding source side is filled up by their translated sentences using a pre-trained target → source translation system. Currey et al. [44] extended the concept by populating parallel sentences by simply copying target monolingual sentences onto the target side. They used these copied techniques and back translation and reported improvement in BLEU score for Turkish ↔ English and Romanian ↔ English low-resource pairs. As opposed to back-translation, a study in [92] proposed *forward translation* to improve NMT by supplementing their training data with synthetic data generated by MT on the target side. Although they only noticed marginal gains, their technique does not require a pre-existing NMT system trained to translate in the other direction.

Other than using monolingual corpus, few of the studies have considered generating new synthetic sentences from original sentences by replacing words or phrases using linguistically motivated rules/dictionaries [232, 56]. Several studies have also exploited comparable corpora to mine parallel segments (sentences and phrases) that are translation equivalents [179]. Parallel sentences extracted from comparable corpora have been long identified as a good source of synthetic data for MT. However, most of the studies rely on large parallel sentences to be trained [71, 194, 25, 31, 195]. As a result, these approaches are not feasible for our case. In addition, comparing each source segment to each target segment leads to prohibitive quadratic time complexity [78, 79]. A typical approach reduces search complexity by aligning documents and then extracting sentences/phrases from within the aligned documents. However, for most low-resource language pairings, such as Manipuri-English, a substantial number of high-quality document-aligned pairs are not accessible.

Although data augmentation techniques are promising, each technique has practical limitations when applied to low-resource language pairs. Back-translation assumes that an MT system exists between the given language pair. Moreover, its performance depends on many factors such as the original-synthetic parallel data ratio, the domain relatedness of the parallel and monolingual data, etc. [58, 60, 102]. Language-specific resources (e.g., bilingual dictionaries, POS taggers, dependency parsers) are required for word or phrase replacement-based augmentation approaches [232, 56], which many low-resource languages lack.

2.2.2 Multilingual MT

Another approach for handling low-resource MT is projecting the translation task as a multilingual problem, refer to as multilingual MT [45]. Multilingual MT models handle translation between more than one language pair. Studies have shown that when the number of languages is limited and if they share similar linguistic characteristics, multilingual models outperform bilingual models [253]. The method is predominantly studied for NMT paradigm [98], apart for a few studies related to SMT [236, 192, 76, 83, 17]. This is primarily because of the NMT model's capacity to learn a shared semantic representation between languages. Multilingual MT aims to develop a single model for translation between multiple languages, including low-resource pairs, by effective use of available linguistic resources. The models are desirable for low-resource MT. The model provides a mechanism to utilize data from high-resource language pairs to improve the translation of low-resource language pairs. Studies related to multilingual NMT can be classified into broad categories: (I) *Transfer learning* (II) *Unseen language pairs MT*.

I. Transfer Learning

As multilingual MT systems consider several languages in the same vector space, it is possible to provide additional translation signals from a high-resource (parent) language pair to improve (child) low-resource MT. Such technique is refer to as *transfer learning* [167]. Over the years, transfer learning has received a lot of attention. Majorities of the previous studies have explored transfer learning on the source side. In this case, high-resource and low-resource languages combination are trained to translate to the same target language using techniques like jointly training [98], meta-learning [69], fine-tuning the parent model with the child's language pair data [253], etc. On the other hand, target-side transfer learning is much more challenging than the source-side. Transfer learning prefers target-language-invariant representations, whereas distinct target languages necessitate target-language-specific representations. Transfer learning's success is dependent on striking the correct balance between this components [98, 46]. Relatedness between the parent and child languages is another critical factor affecting multi-lingual MT performance [252]. Although it is crucial to address the linguistic divergence characteristics issue, surprisingly few works address it [45]. To deal with the lexical divergence between the parent and child languages, authors in [68, 107] initialized the model with pre-trained CLEs. Few studies have tried to explicitly utilize language relatedness by using BPE encodings between the parent and child languages [155], transliteration [142], etc. Rudra Murthy et al. [187] have proven that reordering the parent sentences to reduce the word order divergence between source languages is beneficial for low-resource scenarios. Similarly, Kim et al. [107] mitigate syntactic divergence by training the parent

encoder with noisy source data. This procedure ensures that the encoder is not over-optimized for the parent source language. Gu et al. [68] use a mixture of language expert networks to transfer the syntax-sensitive contextual representations better.

II. Unseen Language Pairs MT

Unseen language pairs MT models assume that even if two languages do not have parallel corpora, they are likely to share one with a third language, referred to as pivot language. *Pivot-based MT* is a special kind of multi-lingual MT that provide a mechanism to develop MT systems for unseen language pairs by exploiting parallel corpora for other language pairs: source-pivot and pivot-target. A simple approach for pivot-based MT is to generate the target sentence by cascading the source sentence via the source-to-pivot and pivot-to-target systems at test time. The approach is independent of the translation technology and can be used with SMT [236], RBMT [244], or NMT [33] systems. It is also applicable to multi-lingual NMT system [125].

Apart from pivot-based translation, *zero-shot NMT* and *zero-resource NMT* approaches are also well studied for translation between unseen language pairs. Multi-lingual NMT allows generating reasonable target-language translations for a source sentence, even if the MT system for the language pair has not been specifically trained [98]. Such translation scenario is referred to as *zero-shot* translation system. Although promising, the performance of standard zero-shot systems is generally inferior to that of the pivot-based translation system [98, 169]. Few studies try to reduce differences between encoder representations in the zero-shot setting to enhance translation results [8, 95]. Ha et al. [74] recommended filtering

the softmax output, forcing the model to translate into the target language, to address the problem of generating words in the wrong language. *Zero-resource NMT* is an enhancement of zero-shot NMT. In this method, the training process takes into account an objective specific to the language pair in question for adapting the system, particularly for the specific language pair [62]. Several methods have been explored to customize the training objective without using any true source-target parallel corpus. Some of the prominent techniques includes synthetic corpus generation [125], iterative training [62], teacher-student training [32], and combining pre-trained encoders and decoders [109], etc.

Although multilingual MT might be a viable alternative to Manipuri-English MT, the strategy would need to pivot languages with comparable linguistic properties to Manipuri. This would require investigating additional languages, which is beyond the scope of this thesis.

2.3 Unsupervised Machine Translation

As discussed in the above sections, even though data augmentation and multilingual NMT techniques have alleviated parallel sentence dependency problems to some extent, they still demand several parallel resources. Such resources are not available for the bulk of the low-resource languages, including Manipuri. To overcome this issue, several authors have explored unsupervised MT (UMT) algorithms that depend solely on the source and target monolingual corpora [127, 14, 224, 42]. Unfortunately, UMT is much more challenging due to the lack of alignment information between source and target languages. Nonetheless, they are promising since the monolingual corpora are usually easy to collect compared

to parallel data. The ability to learn translation features without using expensive parallel sentences will be a massive boost towards the progress of low-resource MT studies. Therefore, this thesis is dedicated to exploring the efficacy of UMT models on the low-resource Manipuri-English language pair.

UMT has its roots from word-based decipherment approaches [114, 181]. These word-based models are later enhanced by incorporating alignment models [54] and heuristic characteristics [152]. Recently, Artetxe et al. [14] and Lample et al. [126] proposed fully-fledged unsupervised MT systems. These unsupervised systems are motivated by the success of unsupervised cross-lingual embeddings [43, 11]. Details regarding unsupervised cross-lingual embeddings are presented in Chapter 5. Unsupervised MT research can be divided into two categories: (i) unsupervised statistical MTs (USMT) and (ii) unsupervised neural MTs (UNMT).

Following the NMT paradigm, UNMT model initialise the encoder-decoder architecture with cross-lingual embeddings [127, 14]. Lample et al. [126] use a single encoder and a single decoder for both the source and target languages. Artetxe et al. [14], on the other hand, utilizes a shared encoder but two independent decoders. The models are then enhanced by using denoising auto-encoder and iterative back-translations. Recently, cross-lingual masked language models (CMLM) [42, 224] have been proposed for effective initialization. MASS [224], a CMLM-based UNMT model, is reported to achieve a BLEU score of 37.5 for English-French outperforming XLM [42] (previous best UNMT model) and attention-based NMT model [16]. These models assume that a pair of sentences/phrases from two different languages can be mapped to a shared-latent space via cross-lingual embeddings [126, 14, 42, 224].

Following the initial work on unsupervised NMT [14, 126], it was argued that

the modular architecture of phrase-based SMT may be more suitable for low-resource MT. Under this motivation, Artetxe et al. [12] and Lample et al. [127] proposed unsupervised SMT model. Following the similar concept for obtaining the initial alignments as the UNMT approach, these models first learn cross-lingual n-gram embeddings from monolingual corpora. Unlike UNMT, these cross-lingual embeddings are then used to generate an initial phrase-table of an SMT model that includes an n-gram language model and a distortion model. The initial system is then fine-tuned via iterative back-translation. These USMT models obtained significant improvements over the previous UNMT systems [14, 126]. A detailed description of USMT and UNMT models is presented in Chapter 7.

Despite all the hype, the efficacy of UMT models is dependent on various factors like source and target language corpora quality, linguistic characteristics, etc., as discussed in Section 1.3. Therefore, the effectiveness of using the off-the-shelf UMT models on the language pair requires a thorough investigation. In Chapter 6, we provide an empirical evaluation of the previous approaches to the Manipuri-English language pair.

2.4 Summary

This chapter provides a detailed review of several data-driven MT techniques. We begin by reviewing essential SMT and NMT, which serve as the foundation for UMT models. After that, we go over the different key solutions for adapting SMT and NMT to low-resource contexts. We have discussed their strength and weaknesses. The final part of this chapter covers the fundamentals of UMT methods and recent trends.

3

Manipuri-English Comparable Corpus

This chapter presents a Manipuri-English comparable corpus to facilitate cross-lingual studies between Manipuri and English. The corpus has been created by collating text from two publicly published news sources on the internet, namely *Sangai Express* and *Poknapham* in Manipur. Almost all Manipuri editions are created using proprietary tools that generate texts in customized non-standard and non-unicode encodings. This chapter also proposes tools to transform the non-unicode text into unicode. All the corpus articles are verified and further aligned

at different sub-corpora levels, namely date, and document.

3.1 Introduction

With the increase in the availability of digital content in different languages on the internet, cross-lingual text processing, and technology development is becoming important research area for various applications such as information retrieval, machine translation, bilingual dictionary induction, etc. However, many such studies need a large volume of a parallel text corpus. As creating a large volume of parallel corpus is an expensive and time-consuming task, it poses many challenges in building such systems for low-resource languages.

In recent studies, researchers have started exploring comparable corpus as an alternative resource to parallel corpus for building various cross-lingual applications [179, 111, 34, 180, 250]. A parallel corpus between two languages consists of document pairs where a document in one language is the translation of another document in another language. Unlike parallel corpus, comparable corpus consists of bilingual texts that are not direct translations but are related to each other based on several degrees of comparability [201, 221]:

1. *Strongly Comparable*: document-aligned bilingual texts that are not an exact translation but share similar theme/idea/time/topic with balance content. News articles in different languages reporting the same event, Wikipedia documents describing the same topic in different languages, etc., are good examples of the strongly comparable corpus.

2. *Weakly Comparable*: bilingual texts that are aligned at the level of sub-corpora according to specific or combinations of criteria like domain, loca-

tion, genre, thematic, etc. Document-level alignment is usually not possible in the case of weakly comparable corpora.

3. *Unrelated text*: these are random texts in multiple languages which can still be used for some comparative linguistic purposes.

As many websites such as Wikipedia, News publications, etc., host quality comparable content, comparable corpora are relatively easier to obtain than parallel texts. Further, based on the study [179], a comparable document pair has the edge over a manually translated parallel document pair as it captures a more natural way of formulating text than manual translation. As a result, usage of comparable corpora for various cross-lingual text processing tasks like Bilingual Dictionary Induction (BDI) [80], Machine Translation (MT) [196], Cross-lingual Information Retrieval (CLIR) [154], etc. has attracted growing interest from the researcher in recent times.

This chapter presents a comparable corpus for Manipuri* and English language pair, by collating publicly available news articles on the Internet from two leading news publications, namely *Sangai Express*† and *Poknapham*‡ with dual editions in English and Manipuri. In regards to the Manipuri-English parallel corpus, it is in a very nascent stage. There are only a few thousand publicly accessible parallel sentences of varying domains [94, 18, 75], which are not sufficient for most the cross-lingual studies. Motivated by this, this chapter presents a large-scale Manipuri-English comparable news corpus. The corpus consists of 5.62 million

*Meitei Mayek is another script used for writing Manipuri. However, as most of the presently available online Manipuri texts are in Bengali, we have considered only texts written using Bengali script (https://en.wikipedia.org/wiki/Meitei_language).

†https://www.thesangaiexpress.com/

‡http://poknapham.in/

Manipuri and 5.79 million English tokens. This is the first effort to create a comparable corpus for the language pair to the best of our knowledge. Further, this corpus is also *date* and *document* aligned by using semi-automated and manual alignment procedure.

Unlike other major languages of India, the Manipuri language poses a unique challenge due to the unavailability (or very limited) of Unicode compatible digitized text. Though the script used for writing Manipuri documents* has corresponding Unicode, the majority of the news publications in the Manipuri language are either in PDF files or non-Unicode text generated using a proprietary encoding scheme. For example, the Manipuri edition of Poknapham publication displays Manipuri text in the local script using a proprietary encoding scheme between Manipuri font and roman character. While generating a compatible Manipuri document, the Roman letter-based encoding texts need to be transformed into Manipuri text in Unicode. In this chapter, we also propose an effective rule-based framework for transforming the encoded Manipuri documents into corresponding Unicode-based Manipuri documents. Furthermore, a systematic analysis of Zipf's and Heaps' law is also provided to understand word frequency distribution across the language pair.

3.2 Related Studies

Despite hosting an abundant amount of comparable data sources for multiple languages, obtaining reliable bilingual texts from the Web is not a trivial task due to its size, inconsistent, unstructured, and heterogeneous nature [221]. Over the years, several methods have been proposed for compiling comparable corpora from

*Bengali script and Meitei Mayek scripts are used for writing Manipuri documents.

various multi-lingual web sources. The methods can be grouped under two broad categories: (1) Website-meta information-based method and (2) content-based method.

Website-meta information-based methods exploit the meta-data like URLs, naming conventions, etc. It is one of the most effective and efficient approaches for generating comparable corpora. A distinct advantage of such methods is efficiency, as simple URL matching does not need to extract the HTML content to find document pairs [221]. For instance, [196] presented a strategy for building Arabic and French languages pair domain-specific comparable corpora. They exploited the categorization and the multilingualism of Wikipedia documents' meta-data to compile a comparable corpus. Following a similar concept, [66] also generated an English-Punjabi comparable corpus from Wikipedia. Other than Wikipedia, the work of [220] described a website-meta information-based method to obtain document-level alignments for several Indian languages from the website, Mann Ki Baat*. [234] build a comparable corpus by exploiting the meta-information of the patent websites. Mining comparable corpora from Wikipedia is generally convenient. There are inter-language links from a Wikipedia page in one language to an equivalent page (describing the same topic) in another language. The BUCC†, a premier workshop series on Building and Using Comparable Corpora, utilizes Wikipedia articles. However, Wikipedia articles are available only for a limited number of languages. Unfortunately, only a limited amount of Manipuri documents are available on Wikipedia. On the other hand, most communities with their native language tend to have news publications in their language. Manipuri

*https://www.pmindia.gov.in/en/mann-ki-baat/
†https://comparable.limsi.fr/bucc2019/bucc-introduction.html

also has news publishers publishing articles in both English and Manipuri. However, comparable corpora generation from news websites is generally problematic as there are no inter-language links [90].

Another widely used approach is the content-based method. In this method, the content of the documents is analyzed with zero assumption about the document structure or meta-information to obtain comparable corpora [3]. For instance, [206] proposed a content-based method to align news articles in Hindi-English language pair. They first identified top news items on news websites by exploiting Google's news feed. Then, similarities between news items are calculated by translating the Hindi articles into English and comparing it with English news articles. Following a similar idea, [220] used independently trained neural MT systems to align documents across 10 Indian Languages crawled from the Press Information Bureau (PIB)*. Instead of directly exploiting inter-language links, [243] presented an approach for aligning Wikipedia's multilingual content by analyzing the co-occurrence of link topology of topics and subtopics between Japanese-English language pairs. Several other content-based methods have also been developed for aligning bilingual documents from crawled websites. However, they require costly features such as n-gram translations [47], phrase-based statistical MT [64], large bilingual dictionary [27], etc. To alleviate the requirement of expensive resources, [188] exploited a topic mapping model to create an English-Arabic comparable corpus. They first extract the topics of both source and target documents using Latent Dirichlet Allocation (LDA) [23]. The source and target language topics are then mapped to obtain a topic dictionary. The dictionary is later utilized in estimating similarities between the documents. Al-

*https://pib.gov.in/indexd.aspx

though the method does not rely on expensive parallel resources, they depend on many pairs of strongly comparable documents to align the topics. Further, they incorporated traditional translation-based features to boost the alignment performance. Content-based methods are more flexible and versatile but, at the same time, more resource-demanding. Unfortunately, to the best of our knowledge, sufficient in-domain bilingual resources are not currently available for Manipuri-English that can facilitate content-based corpora generation between the language pair. Apart from resource requirements, content-based methods generally suffer from scalability issues [202, 196]. Therefore, this study relies on the news website meta-data followed by a manual alignment procedure to generate a comparable corpus.

3.3 Unicode Conversion

As opposed to Unicode, most online Manipuri texts are available in ASCII-based encoding. These are non-standard encoding generally distributed by proprietary distributors. As a result, there is no standardization in defining the number of glyphs* per character. Moreover, the mapping of glyph/glyphs to a Unicode code point† is also not uniform. One requires the specific font to be installed on the local machine to view the correct text. Ultimately, text processing over the text with non-standard encoding is not as convenient as English or any other Unicode encoded text. The typical text processing operations like searching, sorting, etc., are not generalized across fonts, even for the same vocabulary. Therefore, it becomes unavoidable to first convert the Manipuri non-Unicode texts to Unicode.

*It is a primary symbol from an agreed set of symbols that, in single or combined with other glyphs, is intended to represent a character for writing purpose.

†An integer value that uniquely identifies a character.

In this regard, varieties of Unicode font converters have been reported for Indian languages, including Hindi, Oriya, Marathi, Sanskrit, Gujarati, Kannada, Malayalam, Tamil, Bengali, etc. [119, 174]. Majorities of these studies have considered rule-based approaches, as rule-based are generally more convenient than learning-based models. Learning-based models would require training data for each proprietary font, while a carefully designed set of rules is generally sufficient for the conversion [177, 176]. Although Bengali font converters may also be feasible for Manipuri text, previously proposed Bengali converters being rule-based works only on specific fonts [174, 130].

Figure 3.1 shows the proposed rule-based framework to convert the ASCII-based Manipuri texts (especially for the Sangai Express, the Poknapham, and their compatible texts) with the example words রিপোর্টর (reporter) and কম্মিটি (committee) to Unicode. The framework consists of two main components:

1. *Mapping table*: ASCII encoded character/characters are mapped to the corresponding Unicode character/characters using the mapping table.

2. *Dictionary*: The conversion to Manipuri Unicode text is not straightforward due to various complex Unicode transformation rules. A dictionary is used to induce a set of rules specific to the language.

Although the core idea of the methodology (based on a mapping table and a dictionary) is drawn from the earlier studies used for other Indian languages [177, 130], we make several changes to adapt the method for our task. Specifically, unlike the study in [177] that uses the framework to convert various Indian languages to phonetic-based transliteration (IT3), we adapt the framework to transform the ASCII-based Manipuri texts to Unicode. The merging of two or more glyphs on

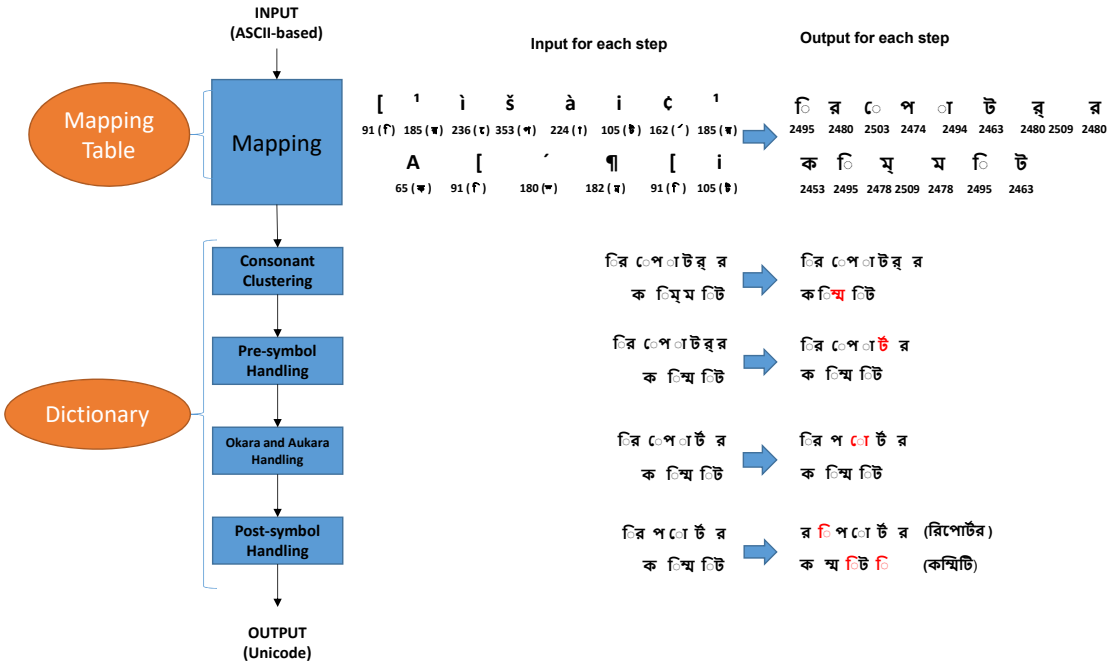


Figure 3.1: Unicode Conversion Framework of Manipuri texts with examples. The number below each character represents the respective code point.

both ASCII and Unicode sides to get the desired character mappings are incorporated in the mapping table itself. For example, in Figure 3.2(a), the character ꯃꯤ is formed by combining multiple ASCII-based characters Ò and ü. Here, the integer at the bottom of each character represents the corresponding code point. Simultaneously, the symbols in the bracket are the glyphs represented by the ASCII-based font if the corresponding font is installed in the local machine. It enables us to resolve many-to-one and many-to-many mapping cases where some ASCII-based character does not represent any ASCII-based glyph, neither mapped to any Unicode character individually, but only works when combined with others are also incorporated in the mapping table itself. The ASCII character with index *161* shown in Figure 3.2(b) is one of such example.

The second major component of the framework is the dictionary. [130] use a

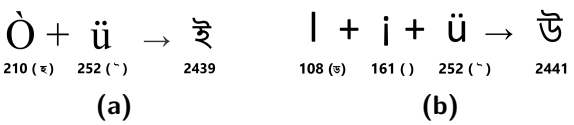


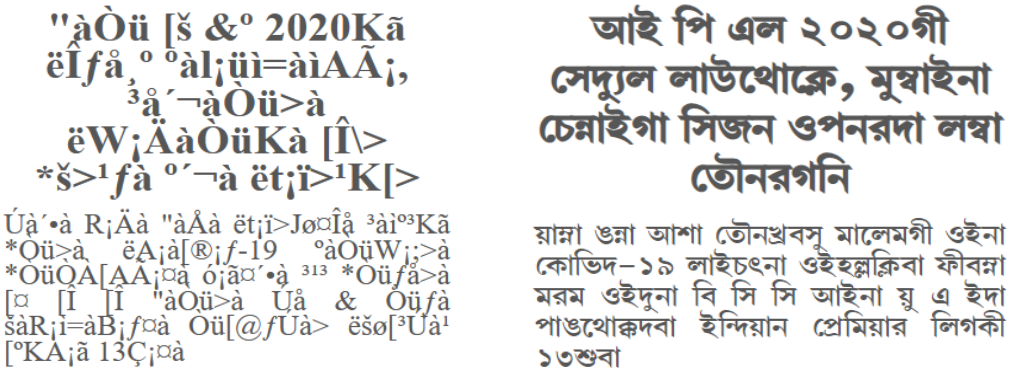
Figure 3.2: Glyphs combination for obtaining a Unicode character.

two-step mapping procedure to convert the non-Unicode text to an intermediate form and then covert the intermediate form to Unicode. As a result, the dictionary (referred to as Unicode transformation rules in the paper [130]) depends on the intermediate form's consistency for all the proprietary fonts. However, in our case, the dictionary is independent of the proprietary fonts encoding. These rules are specifically designed for the language and will remain the same for all the different ASCII-based fonts used for representing Manipuri text. We incorporate the following rules in order:

1. The first rule (*Consonant Clustering*) merges half consonants (except the character র্) with the following full consonant. For example, the half consonant ম্ and full consonant ম are merged to form ম্ম. The clustered symbols from hereon are considered as a single character.

2. The character র্ is a unique character where it must be placed in front of the previous consonant. However, we found that the character র্ appears after the consonant after the mapping process. The second rule (*Pre-symbol Handling*) handles this case by swapping between র্ and the previous consonant.

3. In Manipuri, the dependent vowels can be attached at any position depending on the preceding consonant. In addition, the dependent vowels (ে and ৌ) wrap the consonant. In an ASCII-based rendering system, the posi-



(a) Non-Unicode **(b)** Unicode

Figure 3.3: An article in non-Unicode and its corresponding Unicode encoded texts.

tion of the dependent vowels is as per their attachment positions. Even the dependent vowels (◌ো and ◌ৌ) are rendered as a combination of two separate symbols. However, the dependent vowels must be positioned after the bearing consonant/consonant cluster. The third rule (*Okara and Aukara Handling*) handles the vowels that wrap the consonant/consonant clusters in Manipuri. For example, in the figure 3.1, the character ◌ে is followed by a consonant, then by the character ◌া. Here, we delete the character ◌ে and replace the character ◌া by ◌ো.

4. Finally, the *Post-symbol Handling* rule swaps the position between the dependent vowel and the next character if the dependent vowel is left attached. ◌ি, ◌ে and ◌ৈ are the left-attached vowels in Manipuri.

The Map table and the set of rules are updated and evaluated iteratively, similar to the method presented in [177, 28]. At first, we randomly selected five articles and manually built the initial converter. The table and the rules are modified until all five articles are correctly converted. Then, subsequent batches

Table 3.1: Manipuri-English News Domain Comparable Corpus. The number of sentences presented here have at least three words.

News Website	Language	Documents	Sentences	Words	Vocabulary
Sangai Express	English	6028	77131	2.40M	62914
	Manipuri	5205	71247	1.82M	128187
Poknapham	English	7380	104401	3.39M	80623
	Manipuri	7972	193720	3.80M	218688
Total	English	13408	181532	5.79M	106762
	Manipuri	13177	264967	5.62M	292159

of five documents are converted, and the table and rules are updated iteratively until we obtain zero conversion error. Figure 3.3 presents an article snipped with non-Unicode encoding and its corresponding Unicode converted texts.

The proposed framework is quite efficient and scalable. To convert a word, it requires only four passes over the character sequence of the word. All the ASCII fonts are mapped to the Unicode character(s)/glyph(s) in the first pass. Secondly, the half consonants are merged with the following full consonant. In the third pass, we handled the okara, aukara, and pre-symbol conditions. Finally, we dealt with the post-symbols. Therefore, it takes approximately only O(4n) time to convert an ASCII encoded-word of length n to Unicode.

3.4 Corpus Construction

In this section, we describe the corpus construction process. The corpus is constructed by crawling articles from two of the leading news publishers of Manipur: *Sangai Express* and *Poknapham*. They publish daily news in English and Manipuri. We first visit the news sites' main page and crawl all the news articles by following all the hyperlinks. Both the news publishers provide hyperlinks to all the previously published articles, thereby making it possible to get all the ar-

ticles present in their archives*. While crawling, we save all the URL information to align the documents later. We use python URL handling module [73], the Urllib†, to automate the crawling process. The text content of the articles are then extracted using Beautiful Soup‡ [184], a popular python library for parsing HTML/XML documents. Manipuri texts obtained from the websites are in non-Unicode format. We convert them into Unicode standards using the Manipuri Conversion method discussed above. The documents obtained from the websites are then categorized based on the language, thematic, date, and event to generate a comparable corpus. A detailed description of the categorization procedure is described below.

We first build a domain-aligned comparable corpus by categorizing the crawled articles by language. Specifically, the Sangai Express URL https://www.thesangaiexpress.com/mn/sports/2020/1/1/name-of-the-file.html shows that it is a Manipuri article (given by /mn/)§. In the case of the Poknapham, the English edition is published under a different name *The People's Chronicles*. The English and Manipuri editions have different domain names. Manipuri Edition is registered as http://www.poknapham.in/ while the English edition is published under the domain: http://www.thepeopleschronicle.in/, making it convenient to generate bilingual texts. Table 3.1 shows the detailed description of the domain-aligned comparable corpus. It consists of a total of 13411 English and 13179 Manipuri articles.

On top of the domain-aligned comparable corpus, we further exploit the San-

*This is true at the time of curating the dataset.
†https://docs.python.org/3/library/urllib.html
‡https://beautiful-soup-4.readthedocs.io/en/latest/
§It also provide information about the article thematic (sports), and its publication date.

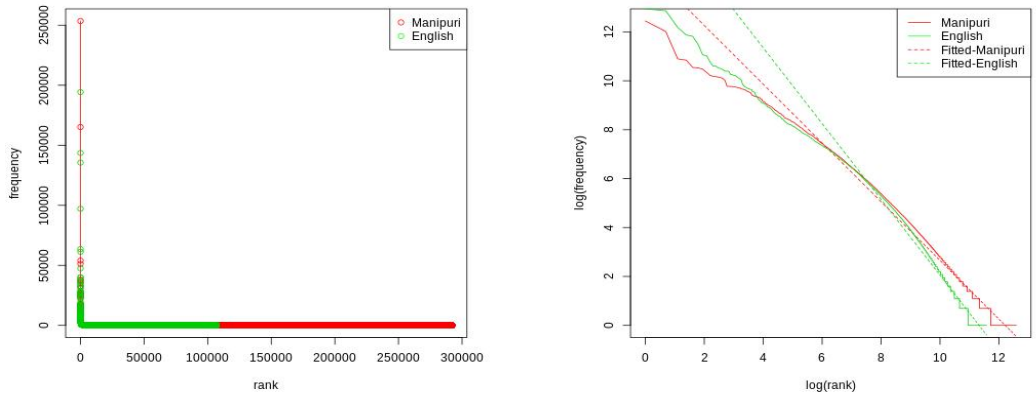
Table 3.2: Manipuri-English Comparable Corpus with stronger degree of comparability

Language	Categories						Total Docs	Total Doc Aligned
	Editorial		Sports		General			
	Total	*Doc Aligned*	*Total*	*Doc Aligned*	*Total*	*Doc Aligned*		
English	246	237	1266	270	4516	2151	6028	2658
Manipuri	239	237	665	270	4301	2151	5205	2658

Figure 3.4: Example of a document-aligned comparable corpus

gai Express URL meta-data to align articles on the level of date of publications. Specifically, an article URL https://www.thesangaiexpress.com/en/sports/2020/01/03/file-name.html shows that it belongs to the *sports* category, and it is published on *2nd Jan 2020**. The articles were published between January 2018 to November 2018. We exploit the date information to build a date-aligned comparable corpus. The Sangai Express publishes news under three different categories (at the time of crawling): (1) General News, (2) Sport News, and (3) Editorial column, in both the English and Manipuri edition. Further, these news categories and date alignments are utilized to find document pairs that describe the same event. We asked two native speakers to check every Sangai Express English and

*The publication date and the date represented in the URLs vary by one day because the news articles on the website are updated a day later after the publication in the respective newspaper.

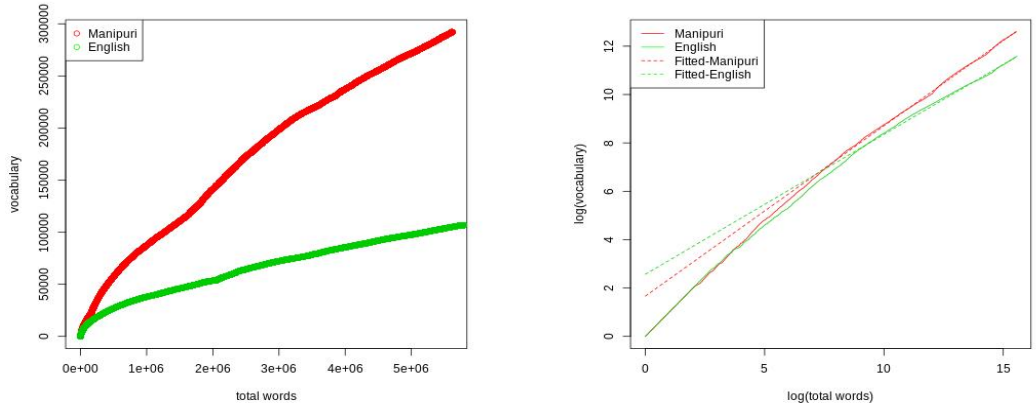


(a) rank vs frequency **(b)** log(rank) vs log(frequency)

Figure 3.5: Zipf's Plot.

Manipuri article pair published for a particular category on a specific date to obtain quality document alignments. The alignment criteria are that the news article pairs must describe the same event, and the two annotators must agree upon it. Table 3.2 shows the detailed description of the document-aligned corpora generated from the Sangai Express. Out of the 6028 documents in English and 5205 documents in Manipuri obtained from the Sangai Express, 2658 document pairs report the same event. Figure 3.4 shows a snipped of a document-aligned document pair; each underlines text with matching color represents a parallel segment. The percentage of document-aligned document pairs (covering almost one-third of the total corpus), and the presence of translations, in terms of sentences and phrases, in each document-aligned pair implicitly shows the potential of the corpus for various cross-lingual studies.

In the case of Poknapham, such meta-information is not available. Therefore, we rely on a simple semi-automated method to get the publication dates. We



(a) total words vs vocabulary **(b)** log(total words) vs log(vocabulary)

Figure 3.6: Heaps' Plot.

observe that the publication date is given as a byline*. We extract the publication dates to generate a date-aligned corpus. The articles were published between August 2018 to June 2020. As the corpus is built using the articles published within the same period by the same publishers discussing similar contents (mainly focusing on regional contents related to the state of Manipur), it shares lots of textual units (sentences and phrases) that are direct translations of each other and should facilitate cross-lingual studies.

3.5 Frequency Distribution Analysis

This section briefly presents the corpus's word distribution using Zipf's and Heaps' laws. Zipf's law [251] states that the rank r of a word and its frequency ($f(r)$), where words are ranked according to their frequencies in the corpus, approximately

*https://en.wikipedia.org/wiki/Byline

follows the power-law relation:

$$f(r) \propto r^{-z} \tag{3.1}$$

Empirical studies in many languages show that z is approximately equal to 1 [137]. In other words, plotting log(r) on the X-axis and log(freq(r)) on the Y-axis can be approximated to a straight line with a slope more or less close to -1, suggesting that frequency decreases very rapidly with rank. Figure 3.5 shows rank on the X-axis versus frequency on the Y-axis plot for both the languages using non-logarithmic and logarithmic scales. The distributions follow that of Zipfian with approximated slopes of -1.2 and -1.55 for Manipuri and English least-squares regression fit (represented as dotted lines on figure 3.5(b)), respectively. The log-log rank-frequency distribution between the two languages is almost identical and roughly shows the empirical law's accurate characterization. However, the predicted slope for the Manipuri is larger as compared to English, showing that the data spareness issue will be more prominent in the case of Manipuri, which we have already seen in our experiment discussed above.

Heaps' law [82, 237] also represents a power-law relation between the vocabulary size (V) and the total number of words in the collection (T):

$$V \propto T^{b} \tag{3.2}$$

where the exponent b is positive and lower than unity showing that the V grows slower than T. Figure 3.6(a) shows the T on X-axis versus V on Y-axis plot for both the languages. Similarly, figure 3.6(b) represents the log(T) versus log(V)

Table 3.3: Corpus Tagset

Tag	Definition
news	Root element.
article	Specify an article.
id	An alpha-numeric string that uniquely identifies a news article.
publisher	Publisher of the article.
document-align	ID of the corresponding document-aligned (reporting same event) article in the other language.
pubDate	Date of publication in YYYY-MM-DD format.
genre	Indicates the genre (General, Editorial, and Sports).
content	Textual content of the news article.

plot. Here, the dotted lines represent the fitted regression line to the relation between $\log(V)$ and $\log(T)$. The results intuitively resemble that of Zipf's law. The slope is on a higher side for Manipuri (slope = 0.70) than for English (slope = 0.58), showing that vocabulary size increases more rapidly for the Manipuri language. This variation is primarily due to the highly agglutinating nature of the Manipuri language.

3.6 Corpus Availability and Format

The documents are tagged using XML format. We maintain one XML file for each language with different tagset as listed in the table 3.3. We plan to make the corpus available publicly to promote cross-lingual studies between Manipuri and English. The template of the annotation is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<news>
<article>
        <id>1-en</id>
        <publisher>Sangai</publisher>
```

```
        <document-align>2-mni</document-align>
        <pubDate>2018-01-03</pubDate>
        <genre>Editorial</genre>
        <content>
        Textual content of the news article.
        </content>
</article>
<article>
    <id>2-en</id>
     ...
</article>
...
</news>
```

3.7 Summary

In this chapter, we present a Manipuri-English comparable corpus. The corpus consists of comparable news articles published from the same publishers within the same period. The articles are also tagged and aligned at the date, and document levels. We believe that this resource will support many researchers in developing various text processing and NLP tools for the low-resource language like Manipuri.

4

Transliteration of English Loanwords and Named-entities to Manipuri

Natural embedding of loanwords from one language to another language has become a common phenomenon in today's writing system. A similar influence is also seen for named-entities. Therefore, it has become critical to develop an effective mechanism for transliterating loanwords and named-entities for several NLP applications. Effective cross-lingual embedding and phrase table construc-

tion are at the core of the majority of the successful unsupervised MT system development. Further, many cross-lingual embedding methods rely on a bilingual dictionary between the source and target language pair. Considering that availability of the digitized bilingual dictionary is still a concern, this thesis exploits transliterated word pairs instead of a bilingual dictionary to establish an inter-language connection between the source and target languages. This chapter addresses the problem of transliterating English loanwords and named-entities to Manipuri. Although machine transliteration research has been ongoing for many years, this essential topic remains untouched for Manipuri-English language pair. Developing a transliteration system may pose many challenges because of the distinct linguistic characteristics between the two languages. In this chapter, we investigate several machine transliteration approaches ranging from (i) dictionary-based mapping and (ii) machine learning techniques, exploiting both the phoneme and grapheme-based representations. This study further proposes a neural hybrid machine transliteration model. The hybrid model alleviates the limitations of individual grapheme and phoneme-based models and enables the model to capture the characteristics of grapheme and phoneme representations simultaneously. Unlike previous hybrid models that rely on linear interpolation or statistical correspondence of grapheme and phoneme sequences, the proposed model is based on the popular neural encoder-decoder based transliteration model. The model strengthens the traditional encoder-decoder transliteration models to a multi-source framework to take advantage of grapheme and phoneme sequences.

4.1 Introduction

Machine Transliteration is the task of automatically converting a word from a source language (or writing system) to a phonetically equivalent word in another target language (or writing system) by conforming to the phonology of the target language. For example, the word *William* in English is transliterated as विलियम in Hindi. Over the past many years, machine transliteration has been considered one of the important sub-problem to assist machine translation [85] and cross-lingual information retrieval [240] by transliterating proper nouns, named-entities, loanwords*, etc. However, for a multilingual and multi-script society like India, transliteration applies beyond just named-entities or loanwords but also at a sentence, paragraph, or document level. It has also become an important problem for many text processing applications. Although transliteration poses no great deal of challenges for the language pairs following similar writing and sound systems, the situation becomes complicated for the language pairs with different alphabets and sound systems, such as English-Manipuri, English-Arabic, English-Hindi, etc. For such language pairs, direct one-to-one mapping from source grapheme to target grapheme may not be applicable. Further, the difference in features, syllables, logographs, and alphabets can be another issue. Other generic challenges like script specifications, phoneme deletion, phoneme insertion, transliteration variants, etc., are also applicable [103]. Although studies related to machine transliteration have been going on for years, this study is the first attempt to transliterate English loanwords and named-entities to Manipuri.

Methods considered for machine transliteration can be broadly classified into

*A word adopted from a foreign language with little or no modification

three main categories, namely, *direct model*, *pivoted model* and *hybrid model*. Direct models [101] are grapheme-based machine transliteration models, where the source language graphemes are directly transliterated into target language graphemes. Whereas in the pivoted models (also known as phoneme-based models), the source graphemes are usually mapped to source language phonemes first, and it is then mapped to target language graphemes [103]. Few studies [113] also consider mapping source language phoneme to target language phoneme first and then mapping to target language graphemes. These models are developed either by adopting either dictionary-based [121, 240] or machine learning-based approaches [67]. However, grapheme or phoneme-based models face the following issues. Pivoted model faces error cascading effect due to involvement of multiple mapping steps. Though grapheme-based models may not face an error cascading effect, it often fails to handle cases when the spelling of the word varies significantly from its pronunciation.

Hybrid models have the potential to overcome these limitations by taking advantage of both the grapheme and phoneme characteristics [165, 153]. In hybrid models, both grapheme and phoneme models are combined using methods like interpolation [20] or grapheme-phoneme correspondence [165]. For example, direct transliteration of the word *cruise* may produce error as its pronunciation differs widely from its spelling. However, as its phoneme representation *(/K/ /R/ /UW/ /Z/)* provides phonetic characteristics, transliteration combining grapheme and phoneme may be more effective. In addition to the limitations mentioned above, grapheme-based approaches need a reasonably large parallel corpus, whereas phoneme-based approaches need a pronunciation dictionary, bilingual pronunciation dictionary, etc. While dealing with resource-poor language pairs (like

English-Manipuri), neither a large collection of parallel corpus nor a quality bilingual pronunciation dictionary may be available. A hybrid framework, which can potentially take advantage of both grapheme characteristics and phoneme characteristics, may be a suitable approach while transliterating from a resource-rich source language to a resource-poor target language.

With the increase in popularity of the neural-based deep learning approaches, the majority of the recent studies [30, 129, 147] on machine transliteration explore deep learning neural techniques. In the Named Entity Transliteration Shared Task conducted as part of The Seventh named-entities Workshop (NEWS 2018) [30], authors in [67] show that the grapheme-based encoder-decoder neural model dominates other models for the majority of the language pairs. Motivated by the above reasons, this chapter proposes a hybrid multi-source encoder-decoder neural model (RNN-based and Transformer-based) that can capture grapheme and phoneme representations characteristics. The effectiveness of the proposed model is then investigated over a resource-poor English-Manipuri language pair for transliterating named-entities and loanwords from English to Manipuri.

To the best of our knowledge, this is the first attempt made for the development of a hybrid encoder-decoder based transliteration model. The proposed hybrid model is an enhancement of the traditional encoder-decoder transliteration model [67, 129] by introducing separate encoders for each source input (phoneme and grapheme sequences, respectively) using multi-source Neural Machine Translation (NMT) techniques. The concept of the multi-source encoder-decoder model has been extensively studied in the NMT paradigm. However, we adapt the multi-source NMT techniques for the transliteration task and explore two of the most widely adopted multi-source encoder-decoder architectures (RNN-based and

Transformer-based). The models are trained to increase the probability of predicting the correct target grapheme sequence given a source grapheme sequence and a phoneme sequence. In addition, this chapter also proposes three novel methods for effectively aggregating the multi-source outputs for feeding to the decoder of the RNN-based models. The proposed aggregation methods better consider the difference in importance between the two source sequences while initializing the decoder.

Experiments on English to Manipuri transliteration on two different resource scenarios: (a) *Moderately Low* and (b) *Extremely Low* resource setting, demonstrate that the proposed hybrid model significantly outperforms its corresponding counterparts. Further, to determine how well our proposed model generalizes across other language pairs having relatively larger training corpus, we also test the proposed hybrid models' performance on four other language pairs: (1) *English-to-Chinese*, (2) *English-to-Thai*, (3) *English-to-Persian*, and (4) *English-to-Hindi*. We also observe that the proposed hybrid transliteration model consistently outperforms its grapheme-based and phoneme-based counterparts for all the language pairs.

Our contributions in this chapter are summarized as follows:

- Comprehensively analyzes the performance of dictionary-based and several machine learning-based techniques by utilizing phoneme-based and grapheme-based representations. It is observed that machine learning-based techniques significantly outperforms dictionary-based models for the language pair.

- Propose a novel multi-source encoder-decoder based hybrid transliteration model that successfully incorporates grapheme and phoneme characteristics.

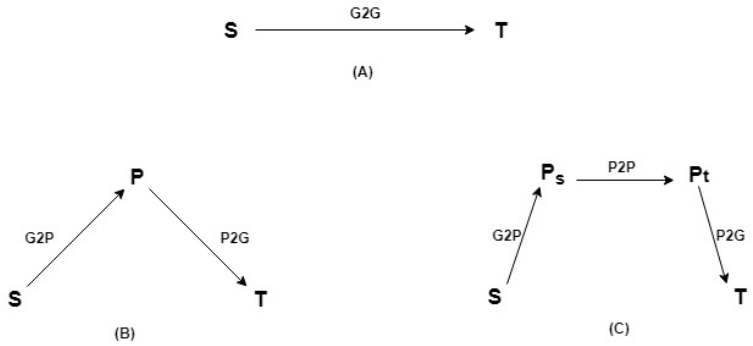


Figure 4.1: A graphical representation of grapheme-based *(A)* and phoneme-based *(B)* and *(C))* approaches where each arrow represents a model (G2G → grapheme-to-grapheme, G2P → grapheme-to-phoneme, P2G → phoneme-to-grapheme and P2P → phoneme-to-phoneme). S and T are the source and target grapheme sequences of word. P, P_s and P_t represent the intermediate phoneme, source phoneme and target phoneme respectively.

- Investigate the performance of the proposed hybrid model modeled on RNN and transformer-based architectures.

- Introduce three new methods for aggregating outputs from multi-source RNN-based encoders namely: (1) *Concatenation*, (2) *Addition* and (3) *Convolution*. The methods are evaluated by comparing with the *Basic* combination method proposed in [252]. It is found that the proposed methods perform relatively better than their counterparts.

4.2 Related Studies

Existing approaches of machine transliteration are broadly classified into *Direct, Pivoted,* and *Hybrid*. Pivoted or phoneme-based approach considers transliteration as a phonetic task and relies on a bilingual pronunciation dictionary for transliteration knowledge. While the grapheme-based or direct approach aims to capture the orthographic mapping between source and target languages using a direct source grapheme to target grapheme conversion model, it ignores the phonetic

level information. Unfortunately, both approaches have their limitations. Pivoted approach [52, 113, 123], as shown in Figure 4.1-(B and C), are more prone to error propagation due to the involvement of multiple modeling steps. Moreover, its dependent on a bilingual pronunciation dictionary is another major drawback. The direct orthographic mapping [61, 178] can avoid some potential errors by eliminating several intermediate phonetic representations, as shown in Figure 4.1-(A). However, it often fails to capture phonetic information when pronunciation differs widely from the spelling [123, 67].

Although studies have shown that the hybrid approach can overcome these limitations by considering characteristics of both the grapheme and phoneme representations [103, 165, 164], very few works have been reported on the development of hybrid models. A primary reason may be the complexity involved in effectively capturing both the grapheme and phoneme characteristics simultaneously. On the other hand, because of the much simpler underlying objective of converting a source sequence to a target sequence for grapheme and phoneme-based models, researchers have explored various modeling methods broadly classified into dictionary-based and machine learning-based approaches. Dictionary-based models requires expensive hand crafted rules to carried out the transliteration process [121, 240, 128]. Instead on relying on expensive rules, several machine learning-based models has also been explored such as statistical framework like maximum entropy [19], expectation-maximization [97], multi-joint sequence model [21, 77], phrase-based machine translation model [156], noisy channel model [146, 99, 240], etc. Until recently, grapheme-based transliteration methods based on phrase-based machine translation [116] was one of the best performing model [61, 178].

With the recent advancement of neural network techniques [38, 16, 238], researchers have also explored various neural network models for transliteration task [185, 120, 129]. Majority of the studies have considered encoder-decoder models [147, 128, 129]. As per the report in most recent Named Entity Transliteration Shared Task [30], as a part of Seventh Named Entity Workshop (NEWS) 2018, most of the participants have considered neural-based models [67, 128, 120, 6], apart from an exception [219] that uses statistical model. It is found that the grapheme-based RNN encoder-decoder model used in [67] consistently outperforms other models in most of the language pairs.

Although several methods have been explored for grapheme and phoneme-based transliteration models, only a few studies are reported for hybrid models in the literature. Researchers in the studies [4, 5, 20] propose hybrid transliteration models where phoneme-based and grapheme-based models built using either the WFSTs (weighted finite-state transducers) [113] or source-channel model [5] are combined using linear interpolation. However, their results show that the linear interpolation method fails to take advantage of grapheme and phoneme information. Study in [4] reports a decline of 3.7% in accuracy as compare to grapheme-based approach. Similarly, the authors in [20] report a decrease in accuracy from 38.7% to 38.0% while comparing with its phoneme-based counterpart. Considering the drawback of linear interpolation methods, authors in [165] propose a model which dynamically incorporates correspondence between graphemes and phonemes representations using three machine-learning algorithms (maximum entropy model, decision tree, and memory-based learning). They achieve an improvement of about 15 to 41% in English-to-Korean transliteration and about 16 to 44% in English-to-Japanese transliteration tasks compared to other models. Going in

line with the dynamic correspondence estimation, authors in [89] present a hybrid approach to the English-Korean transliteration task based on the Statistical Machine Translation framework (MOSES) [115] by enabling factored translation features. Most recent study [153] presents phonology augmented statistical framework for transliteration. They have tested their system on English-to-Cantonese and English-to-Vietnamese pairs and have shown that their proposed method outperforms the grapheme-based counterpart by 44.68%. Similarly, [1] performed a rule-based phonetic rectification prior to grapheme-based mappings. Interestingly, all the above hybrid models follow statistical or rule-based methods. To the best of our knowledge, this study is the first work to combine grapheme and phoneme characteristics for the transliteration task using a state-of-the-art multi-source neural encoder-decoder model.

4.3 Language Transliteration in regards to Manipuri Language

Characteristics of the sound and writing system of the Manipuri language are quite different from that of English. For instance, there are 39 phonemes* to represent 26 graphemes in English, while the Bengali script uses 55 graphemes to represent 38 phonemes in Manipuri language [205]. Further, English is phonetic, and Manipuri is syllabic. As a result of all such differences between the two languages, there are lots of ambiguities associated with the transliteration task between English and Manipuri [123]. Some of the major issues in mapping an English grapheme to a Manipuri grapheme are the presence of one-to-many and many-to-one grapheme maps. Table 4.1 shows some of the mapping ambiguities for English to Manipuri transliteration task. For example, source grapheme ***a*** is

*http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Table 4.1: Examples of one-to-many, many-to-one and many-to-many mapping ambiguities associated with English to Manipuri transliteration task.

One-to-many Mappings			
English	*Manipuri Transliteration*	*Mapping*	*Source Phoneme*
administerial	এডমিনিসট্রেরিয়েল	**a**dministerial → এ	/AE/
sofa	সোফা	sof**a**→ া	/AH/
seaboard	সিবোর্ড	seabo**a**rd → ো	/AO/
almond	অলমোন্দ	**a**lmond → অ	/AA/
Many-to-one Mappings			
English	*Manipuri Transliteration*	*Mapping*	*Source Phoneme*
european	ইউরোপিয়ান	europ<u>e</u>an → ি	/IH/
kohima	কোহিমা	koh<u>i</u>ma → ি	/IH/
darjeeling	দার্জিলিং	darj<u>ee</u>ling → ি	/IH/
Many-to-many Mappings			
English	*Manipuri Transliteration*	*Mapping*	
scientific	সাইন্টিফিক	<u>scie</u>ntific → সাই	
software	সোফটৱেয়র	soft<u>ware</u> → ৱেয়র	
airport	এয়ারপোর্ট	<u>air</u>port → এয়ার	

mapped to different graphemes অ, া , এ , ো , etc. in Manipuri. Similarly, different English graphemes like *e* in *Europ<u>e</u>an*, *i* in *Koh<u>i</u>ma*, *ee* in *Darj<u>ee</u>ling*, etc. can all be mapped to single Manipuri grapheme ি. There are also cases where English grapheme sequences are mapped to Manipuri grapheme sequences using many-to-many mappings, as shown in Table 4.1. Similar cases are also valid for source phoneme to target grapheme mappings in phoneme based models. The source phoneme /AH/ can be mapped to different target graphemes. For example, া in বিমা (bim<u>a</u>) and ো in অলমোন্দ (alm<u>o</u>nd).

Although this is the first effort for transliterating English loanwords and named-entities to Manipur, a few studies work on transliterating Manipuri text written in Bengali Script to Meitei Mayek using rule-based methods[162, 208]. Considering that Manipuri text (same language) written in Bengali script and

Independent vowels				
অ = /a/	আ = /aa/	ই = /i/	ঈ = /ii/	উ = /u/
ঊ = /u/	এ = /ae/	ঐ = /ai/	ও =/o/	ঔ = /au/
Dependent vowels				
া = /aa/	ি = /i/	ী = /uu/	ু = /u/	ূ = /uu/
ে = /ae/	ৈ = /ai/	ো = /o/	ৌ = /au/	
Full consonants				
ক = /ka/	খ = /kha/	গ = /ga/	ঘ = /gha/	ঙ = /nga/
চ =/ca/	ছ =/cha/	জ = /ja/	ঝ = /jha/	ঞ = /nza/
ট = ত = /ta/	ঠ = থ = /tha/	ড = দ = /da/	ঢ = ধ = /dha/	ণ = ন = /na/
প = /pa/	ফ = /pha/	ব = /ba/	ভ = /bha/	ম = /ma/
য় = /ya/	র = /ra/	ল = /la/	ৱ = /wa/	ক্ষ= /kq/
শ = /sha/	ষ = /sxa/	স = /sa/	হ = /ha/	
Pure consonants				
ক্ = /k/	খ্ = /kh/	গ্ = /g/	ঘ্ = /gh/	ং = ঙ্ = /ng/
চ্ =/c/	ছ্ =/ch/	জ্ = /j/	ঝ্ = /jh/	ঞ্ = /nz/
ট্ = ত্ = /t/	ঠ্ = থ্ = /th/	ড্ = দ্ = /d/	ঢ্ = ধ্ = /dh/	ণ্ = ন্ = /n/
প্ = /p/	ফ্ = /ph/	ব্ = /b/	ভ্ = /bh/	ম্ = /m/
য়্ = /y/	র্ = /r/	ল্ = /l/	ৱ্ = /w/	ৃ = /rq/
শ্ = /sh/	ষ্ = /sx/	স্ = /s/	হ্ = /h/	ঁ = /mq/

Figure 4.2: Manipuri phoneme mapping table

Meitei Mayek are orthogonally similar, obtaining such rules is simpler. Whereas constructing rules between orthogonally dissimilar languages like English and Manipuri is an expensive task.

4.4 Dictionary-based approaches

This section discusses the dictionary-based methods (phoneme-based and grapheme-based approaches) deployed for our transliteration task.

4.4.1 Phoneme-based approach

The proposed phoneme-based approach for transliterating English words to Manipuri follows the setup presented in Figure 4.1 (B). Given an English word, it is first transformed the word into a sequence of phonemes of the target language us-

e = এ = ে	y = য়	j = জ	
a = অ = া	l = ল	n = ন	ph = ফ
ei = ঐ = ৈ	gh = ঘ	p = প	ng = ং
ai = ঐ = ৈ	ch = ছ	k = ক	s = স
au = ঔ = ৌ	v = ভ	f = ফ	h = হ
i = ই = ি	w = ব	b = ব	x = স
u = উ = ু	kh = খ	m = ম	z = জ
o = ও = ো	g = গ	r = র	t = ত
jh = ঝ	q = ং	bh = ভ	d =দ
th = থ	c =চ	sh = শ	dh = ধ

Figure 4.3: Manipuri grapheme mapping table

ing a pre-defined dictionary. The intermediate phoneme sequence is then mapped to the corresponding grapheme sequence of the target language to obtain the target transliteration. In this approach, generating a cross-lingual phoneme dictionary is an expensive operation. In an earlier paper [189], a method for adapting the CMU pronunciation dictionary from English to Manipuri is proposed using sequence labeling methods such as CRF and obtaining an F-score measure of 0.991 for phone-level classification and 0.93 word level accuracy. Considering the high accuracy reported in the paper, we consider the modified CMU dictionary proposed in [189] for generating English to Manipuri phoneme-based mapping.

For generating the target grapheme from the intermediate phoneme representation, we use the phoneme-to-grapheme maps for Manipuri language in Bengali scripts developed in the study [204]. This mapping is shown in Figure 4.2. Since a sequence of a vowel following a consonant form a cluster element in Manipuri writing (true for most of the Indian languages), we further apply the following rules while converting from the phoneme to grapheme through Figure 4.2. We map a vowel phoneme to a dependent vowel grapheme when it follows a consonant

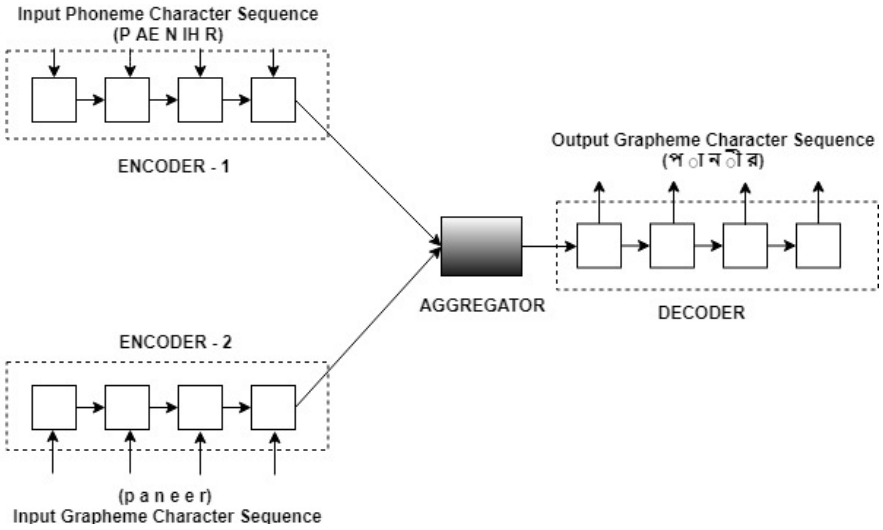


Figure 4.4: Schematic diagram of the proposed multi-source RNN-based Encoder-decoder Hybrid Transliteration Model. This receives two different source sequences through two encoders (ENCODER - 1 and ENCODER -2). The aggregation methods combine the output of the two encoders. The output of the aggregation function is then passed to the decoder layer.

phoneme. Similarly, when a vowel phoneme /a/ follows a consonant phoneme, the consonant phoneme is mapped to a full consonant grapheme.

4.4.2 Grapheme-based approach

Figure 4.3 shows the grapheme-grapheme mapping table between English to Manipuri. Given an English word, corresponding Manipuri transliteration is generated using this table. For example, a named-entity word *Kohima* is mapped as কোহিমা using the considered mapping table in our grapheme-based approach. As discuss in phoneme based transformation above, a vowel is converted to dependent vowel when it follows after a consonant.

4.5 Multi-Source Encoder-Decoder Machine Transliteration Model

The proposed neural multi-source encoder-decoder model transforms the transliteration task as a sequence-to-sequence (seq2seq) problem [38, 37]. We consider one of the most widely adopted seq2seq model, i.e., the `encoder-decoder` [16, 238]. The model consists of two important components: *encoder* and *decoder*. Among the different architectures that follow the encoder-decoder paradigm, the RNN-based [16] and Transformer [238] has been firmly established as the state-of-the-art approaches for various tasks like machine translation, text summarization, image captioning [135, 173], etc. The RNN-based grapheme model is the best performing system in the Named Entity Transliteration task 2018 [30]. Motivated by their success in various seq2seq generation tasks, in this thesis, we explore both the RNN-based and transformer-based architectures to model our proposed multi-source hybrid transliteration.

4.5.1 Multi-source RNN-based model

In RNN-based transliteration model, both the *encoder* and *decoder* are Recurrent Neural Networks (RNNs) connected together. The task of an encoder is to understand the input sequence $x_1, x_2, x_3, ..., x_n$ (source word grapheme or phoneme sequence) and generate the output sequence $y_1, y_2, y_3, ..., y_m$ (target word grapheme sequence). The models follow the same architectural setup discussed for NMT in Chapter 2.1.2. However, for training the model, instead of using sentence pairs, we feed the model with a set of transliteration pairs T containing a list of source word input sequence and the corresponding target word output sequence.

We extend the basic RNN-based encoder-decoder framework with multiple en-

coders to enable the proposed model to learn from multiple input sequences. The multi-source encoder-decoder architecture was first introduced for Machine Translation [252, 160, 134]. We adapt this model for the machine transliteration to incorporate grapheme and phoneme characteristics. Figure 4.4 shows our proposed multi-source encoder-decoder model. Multiple input sequences, i.e., phoneme sequence and grapheme sequence of input word in the source language, are fed into two different encoders (phoneme sequence on ENCODER -1 and grapheme sequence on ENCODER -2). The outputs from the encoders are then merged using an aggregation function (discussed in Section 4.5.1). The output of the aggregation function is then fed to the decoder to obtain the target word grapheme sequence. The purpose of using a multi-encoder is to exploit the characteristics of both the source phoneme and grapheme representations while predicting the target grapheme. Let T_m be the set of tuple $< s_g, s_p, t_g >$ where s_g and s_p are the grapheme sequence and phoneme sequence of the source word, and t_g is the corresponding grapheme sequence of the target word. The objective function of the proposed model is to minimize the following cross-entropy loss:

$$J_m = \sum_{<s_g, s_p, t_g> \in T_m} -logp(t_g|s_g, s_p) \tag{4.1}$$

I. Aggregation Methods

As each input sequence (phoneme and grapheme sequence) of the source word is given through two separate encoders, the representations obtained from the two encoders are different. In this case, the grapheme representation is generally more robust, and we might want to consider the phoneme representation only

when the word pronunciation varies significantly from its spelling. Therefore, a proper aggregation of the two different representations is essential before sending it to the decoder. However, previous approaches do not explicitly model the roles of the individual source sequences. Apart from the two aggregation methods (Basic and Child-Sum Method) proposed in [252], majorities of the studies are dedicated to improving the multi-source attention mechanism [252, 134]. However, we believe that in RNN architecture, proper initialization of the decoder's hidden state will impact the overall performance. In this chapter, we propose three different aggregation methods that take into account the different importance of the individual source sequences and compare them with the method proposed in [252] (This method is referred to as MSHy-Basic from here on).

Let $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ be the encoded representations obtained from ENCODER-1 and ENCODER-2 respectively, as shown in Figure 4.4. This two representations are aggregated using the proposed aggregation functions. In case of LSTM based frameworks, if $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are the cell states of the ENCODER-1 and ENCODER-2 at the end of the input sequence, then the decoder cell state is initialized by element wise addition (+) i.e., $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$. The proposed aggregation methods are described below.

1. **MSHy-Basic :** This method proposed in [252, 160] applies a single linear transformation $\boldsymbol{W}_c$ (parameter matrix) on the concatenation of $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ and then a *tanh* activation function as shown in equation 4.2.

$$\boldsymbol{h} = tanh(\boldsymbol{W}_c[\boldsymbol{h}_1 : \boldsymbol{h}_2]) \tag{4.2}$$

2. **MSHy-Concatenation :** In this method, a *tanh* activation is applied to a

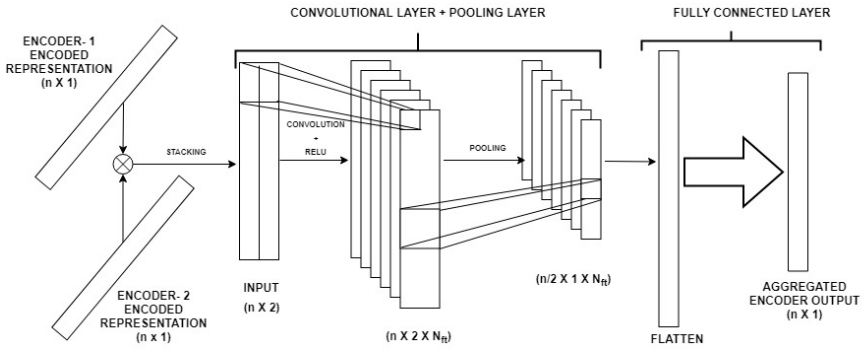


Figure 4.5: Model Architecture of the aggregation function using Convolutional Neural Network.

straight forward concatenation of the encoders hidden state for initializing the decoder hidden state. It is given by:

$$\boldsymbol{h} = tanh([\boldsymbol{h}_1 : \boldsymbol{h}_2]) \tag{4.3}$$

3. **MSHy-Addition :** The third aggregation method is inspired by the additive attention in [16]. In contrast to the MSHy-Basic, each encoder hidden states ($\boldsymbol{h}_1$ and $\boldsymbol{h}_2$) are passed through linear transformations with $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ respectively, where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are parameter matrices to better account for the difference in characteristics of the source sequences. They are then additively combined as follows:

$$\boldsymbol{h} = tanh(\boldsymbol{W}_1\boldsymbol{h}_1 + \boldsymbol{W}_2\boldsymbol{h}_2) \tag{4.4}$$

4. **MSHy-Convolution:**

Our final aggregation method is inspired by the Convolutional Neural Network (CNN) [117]. Here, we use a CNN to aggregate the encoders hidden states $\boldsymbol{h}_1, \boldsymbol{h}_2 \in R^n$, where n is the number of hidden unit of the encoder. A

detail model architecture is shown in the figure 4.5. Input to our CNN is the matrix ($\boldsymbol{H}$) obtained by stacking $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ given by:

$$\boldsymbol{H} = [\boldsymbol{h}_1 \otimes \boldsymbol{h}_2] \tag{4.5}$$

such that $\boldsymbol{H} \in R^{n \times 2}$ and $\otimes$ is the stacking operator. Then, the aggregated encoders output ($\boldsymbol{h}$) is obtained as follows. Firstly, a convolution operation is applied to $\boldsymbol{H}$ using a filter $\boldsymbol{L} \in R^{2 \times 2}$ with *same* padding option and *relu* activation function to obtain a convolution output $\boldsymbol{C} \in R^{n \times 2}$. Secondly, a max pooling is performed on the convolution output $\boldsymbol{C}$ using a pool size of 2×2 to obtain a pooled vector. Here, we have used N_{ft} number of filters to obtained N_{ft} number of convolution matrices and corresponding N_{ft} number of pooled vectors. Finally, a fully connected dense network is used for obtaining the aggregated encoders output $\boldsymbol{h}$ from the flattened pooled vectors. For all of our experiments, we have considered 64 filters (i.e., $N_{ft} = 64$).

II. Multi-Encoder Attention

Our multi-encoder attention is modeled over the global attention mechanism proposed in [140] to look at both the source encoders simultaneously as presented in [160]. Apart from the concatenation method in [252, 160], there are also different ways for computing the attention (flat and hierarchical), as proposed in [134]. As reported in [134] for Multimodal Translation and Automatic Post-editing tasks, the majority of the methods provide comparable performance. Therefore, in this study, we consider a concatenation-based attention method for the proposed RNN-

based multi-source models. Suppose at time step t, the two encoders produce the context vectors $\boldsymbol{c}_t^1$ and $\boldsymbol{c}_t^2$. Then, they are concatenated together along with the corresponding decoder hidden state $\boldsymbol{h}_t$ to compute the final decoder state vector $\boldsymbol{h}_t^d$ using the parameter matrix $\boldsymbol{W}_a$ and *tanh* activation function, which is defined as below.

$$\boldsymbol{h}_t^d = tanh(\boldsymbol{W}_a[\boldsymbol{h}_t : \boldsymbol{c}_t^1 : \boldsymbol{c}_t^2]) \tag{4.6}$$

As computation of context vectors ($\boldsymbol{c}_t^1$ and $\boldsymbol{c}_t^2$) for both encoders follows the same procedure, we represent them as $\boldsymbol{c}_t$ and it is computed by using the same technique as proposed in [140]. First we compute the attention weights (α_{ts}) using the current decoder hidden state and all the encoder states $\bar{\boldsymbol{h}}_s$ as:

$$\alpha_{ts} = \frac{exp(score(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s))}{\sum_{s'=1}^{S} exp(score(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s))} \tag{4.7}$$

$$score(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \boldsymbol{V}_a^T tanh(\boldsymbol{W}_x \boldsymbol{h}_t + \boldsymbol{W}_y \bar{\boldsymbol{h}}_s) \tag{4.8}$$

where S be the length of the source sequence, and $\boldsymbol{V}_a$ be a vector that serves as a fully connected dense layer. $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ are parameter matrices. Then, the context vector is computed as the weighted average of all the encoder hidden states based on the attention weights as:

$$\boldsymbol{c}_t = \sum \alpha_{ts} \bar{\boldsymbol{h}}_s \tag{4.9}$$

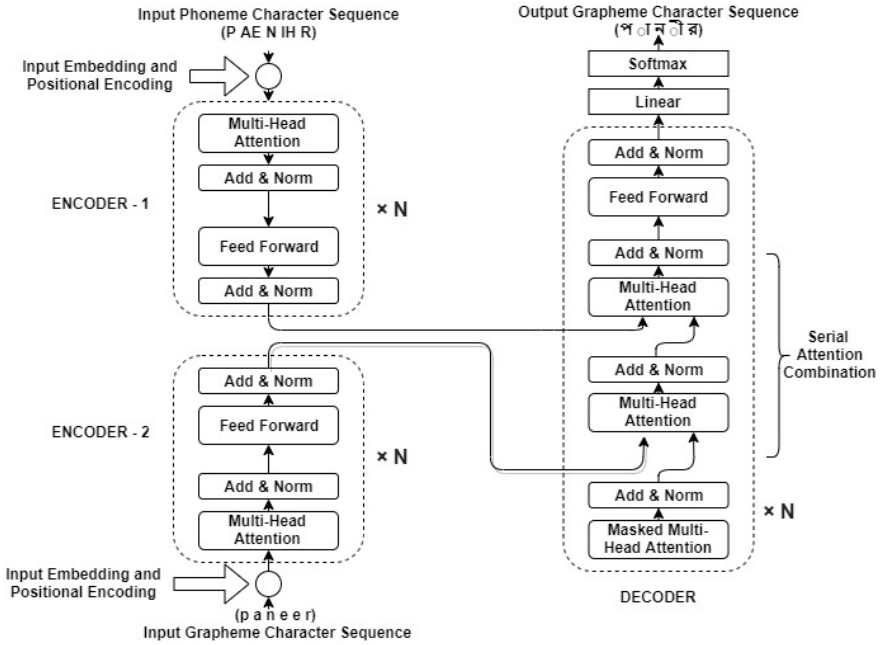


Figure 4.6: Schematic diagram of the proposed multi-source transformer-based Encoder-decoder Hybrid Transliteration Model with Serial Attention Combination.

4.5.2 Multi-source Transformer-based Model

Figure 4.6 shows the proposed transformer-based hybrid transliteration model. Similar to the multi-source RNN-based model, the proposed transformer-based hybrid transliteration model *(MSHy-Serial)* use two encoders (ENCODER-1 and ENCODER-2) to encode the phoneme and grapheme sequence characteristics separately. The architectural design of each encoder is the same as the conventional transformer encoder (discussed in Chapter 2.1.2). However, the structure of the decoder is modified to capture multi-source inputs. In literature, various forms of multi-source transformer models are proposed [135]. As observed in the above paper, different multi-source transformer variants like serial, parallel, flat, and hierarchical provide comparable performances. Considering these observations, we have considered the serial model to combine grapheme and phoneme sequences in

the proposed hybrid model, as shown in Figure 4.6.

The serial attention combination computes the encoder-decoder attention serially for each encoder. The query set for the first encoder-decoder attention layer is the set of the context vectors obtained from the preceding decoder attention layer. However, the key and value sets are obtained from one of the source encoders (from ENCODER-2 in our case). The query set of the subsequent encoder-decoder attention is the output of the preceding sub-layer, while the key and value set is obtained from the other encoder (ENCODER-1). Similar to other sub-layers, these encoder-decoder attention sub-layers are interconnected with residual connections.

4.6 Experimental Setup

To evaluate the performances of the proposed hybrid models, we extensively investigate the models on English-Manipuri transliteration on two different resource scenarios. A detailed description of the experimental setups are presented below.

4.6.1 Dataset

This study considers an English-to-Manipuri parallel corpus publicly available at http://tdil-dc.in. This dataset is originally distributed for building Manipuri machine translation system in the Tourism Domain. It consists of 9892 parallel sentences. From this corpus, a moderate size transliteration dataset consisting of 6035 transliterated words is manually extracted for evaluating the models. It includes a total of 3402 named-entities and 2633 English loanwords. Some of these named-entities are in plural form. Such plural forms are first manually stemmed to

Table 4.2: English-Manipuri Language Pair Dataset Description

	Moderately Low	*Extremely Low*
Training	4428	2000
Development	1000	500
Testing	607	607

the corresponding singular form. For example, the plural word ꯍꯩꯅꯣꯡꯁꯤꯡ*(mangoes) is stemmed to its root word ꯍꯩꯅꯣꯡ†(mango) by removing the plural morphological inflection ꯁꯤꯡ. Similarly, the English word *mangoes* is stemmed to *mango*. The processed dataset is then randomly split into three, i.e., training, development, and testing sets, as shown in the *second* column of Table 4.2. However, in forward transliteration‡, most of the words in the source language are often associated with a list of spelling variants in the target language. To capture such variations, we also populated our testing set by manually adding all the spelling variants in Manipuri for each English word in the test set. For example, the word *botanist* can be written as either ꯕꯣꯇꯥꯅꯤꯁ꯭ꯠ or ꯕꯣꯇꯥꯅꯤꯁꯦꯠ in Manipuri. So, we have added both the variants of *botanist* as its reference transliterations. Our expanded testing set contains, on average, 3.21 reference transliterations for each English word. We further investigate the model performance on limited English-Manipuri training data to determine the model's capability to adapt to the extremely low-resource setting. To simulate the extremely low-resource scenario for the language pair, we randomly select 2k transliterated word pairs as training data along with 500 pairs as the development set from the original training data discussed above. The extremely low resource dataset is presented in the *third* column of table 4.2.

*Transliteration in Roman alphabet: heinousing

†Transliteration in Roman alphabet: heinou

‡Transliteration of a word from its original language to a foreign language.

4.6.2 Evaluation

In this study, we consider two standard evaluation metrics, namely, *word accuracy* and *character accuracy*, for evaluating the performances of different models.

- **Word Accuracy (WA)** : Word accuracy measures the correctness of the predicted transliterations produced by the system and is one of the evaluation metrics used in NEWS 2018 [30]. If N is the number of source words in test set, $r_{i,j}$ is the j^{th} reference transliteration for i^{th} word in the test set and t_i is the predicted transliteration of i^{th} word. Then, WA is given by:

$$WA = \frac{1}{N}\sum_{i=1}^{N}\left\{\begin{array}{c} 1, \ if \ \exists r_{i,j} : r_{i,j} = t_i \\ 0, \ otherwise \end{array}\right\} \times 100 \tag{4.10}$$

- **Character Accuracy (CA**) : It measures the number of character insertions, deletions, and substitutions between the predicted transliterated word *(P)* with original transliteration *(T)*. It is based on edit distance algorithm [132] and is given by:

$$CA = \frac{len - ED(P,T)}{len} \times 100 \tag{4.11}$$

where, *len* is the length of predicted *(P)* or original *(T)* word whichever is larger and ED gives the edit distance between P and T. In this study, we have reported the best CA obtained from among the reference transliterations.

4.6.3 The Model Configurations

I. RNN-based models

As the performance of an RNN-based encoder-decoder model depends on its architectural framework, we evaluate the proposed hybrid model using three widely adopted standard frameworks:

- **Single Layer Encoder** : A single layer uni-directional RNN (Recurrent Neural Network) [38] for each encoder with a single layer RNN decoder.
- **Stacked Uni-directional Encoder** : Two layers stacked uni-directional RNN on each encoder combined with a single layer RNN on decoder.
- **Bi-directional Encoder** : Motivated by the success of bidirectional RNN models [16], our third framework consist of a single layer bi-directional RNN on each encoder combined with a single layer RNN on decoder.

For each of the frameworks, we have considered both widely adopted recurrent neural network cells: (1) Long Short-Term Memory (LSTM) [88] (2) Gated Recurrent Unit (GRU) [41]. The experimental RNN-based models are implemented using TensorFlow seq2seq* model. We use Adam optimizer [110] with a learning rate of 0.001 and batch size of 32. Both the recurrent dropout and regular dropout are set to 0.2. The size of the hidden layer of the RNN decoder is fixed to 512 and the embedding dimension to 256.

*https://www.tensorflow.org/tutorials/text/nmt_with_attention

II. Transformer-based Models

We modify the tensorflow implementation of the transformer* to implement the proposed hybrid transformer-based model. The model parameters are optimised using Adam optimizer [110] with initial learning rate 0.2. The Noam learning rate decay are set to $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 10^9$, and 4,000 warm-up steps [238]. We set the dropout to 0.1. In the multi-head attention layer, we use 8 heads. To make the transformer-based models comparable to RNN-based models, other hyper-parameters are kept relatively similar to RNN-based settings. The batch size is fixed at 32, and the dimension of the hidden layer in the feed-forward layer to 512 for all the experiments. We test the models on two different model dimensions (256 and 512) and also investigate three different encoder and decoder layer settings (2-layer, 4-layer, and 6-layer).

4.6.4 Generating English Phoneme Representation

We consider a publicly available English grapheme-to-phoneme (G2P) conversion toolkit† to generate phoneme representation. The choice is also motivated by our preliminary experimental results (presented in Section 4.7.1). We observe that the pronunciation dictionary presented in [189] is not very effective in capturing the pronunciation of English named-entities and loanwords. The G2P toolkit is trained on CMU English pronunciation dictionary using a 3-layer transformer model [238] with 256 hidden units. It gives a Word Error Rate (WER) of 20.6% on CMU dictionary datasets as compared to WER of 24.4% using the standard

*https://www.tensorflow.org/tutorials/text/transformer
†https://github.com/cmusphinx/g2p-seq2seq

Table 4.3: Preliminary experiment results comparing dictionary-based approaches with several machine learning based transliteration models in word accuracy and character accuracy.

	Dictionary		*LSTM*		*GRU*		*Transformer*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	12.42	60.23	24.17	76.38	25.06	76.99	23.54	75.23
Grapheme	20.89	72.62	64.36	91.12	67.00	91.34	66.01	91.17

WFST-based Phonetisaurus G2P toolkit*.

4.7 Results and Discussions

4.7.1 Preliminary Experiments

We first perform a preliminary investigation to determine the performances of dictionary-based approaches presented in Section 4.4. The dictionary-based models are compared with several machine learning-based approaches. Specifically, neural-based seq2seq models (LSTM, GRU, and Transformer) trained using a moderately-low dataset (refer Table 4.2). We consider the single-layer encoder setting for LSTM and GRU. For transformer, we consider 2-layer encoder and decoder layer setting (model dimension = 256). Other parameters are kept the same as described above. To make the systems comparable, the phoneme representations of the seq2seq models are also generated using the modified CMU dictionary presented in [189]. Table 4.3 shows the experimental results in terms of CA and WA for both the phoneme and grapheme. It is evident from the results that for all the cases, the learning-based models significantly outperform the dictionary-based models for the language pair. This shows the ineffectiveness of the dictionary mappings in solving all the ambiguities associated with the task (refer Section 4.3). These results also justify the choice of the LSTM, GRU, and

*https://github.com/AdolfVonKleist/Phonetisaurus

Table 4.4: Performance of different RNN-based transliteration model in word accuracy and character accuracy on English-Manipuri language pair. *LSTM/GRU* stands for single layer encoder framework with LSTM/GRU cell, Similarly, *Stack-LSTM/Stack-GRU* stands for Stacked Uni-directional encoder framework with LSTM/GRU cell and *BiLSTM/BiGRU* for Bi-directional encoder framwork with LSTM/GRU cell.

A: Results on Moderately Low Resource Scenario												
Model	*LSTM*		*GRU*		*Stack-LSTM*		*Stack-GRU*		*BiLSTM*		*BiGRU*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	40.76	83.21	41.91	82.88	41.41	83.63	43.56	84.02	42.74	83.46	45.87	84.8
Grapheme	64.36	91.12	67.00	91.34	65.51	91.00	71.29	92.72	70.62	91.81	73.27	92.66
MSHy-Basic	66.17	90.68	68.65	92.06	67.99	91.54	72.61	92.92	71.45	92.7	77.72	93.72
MSHy-Concatenation	65.68	91.00	**71.62**	**92.62**	69.14	92.74	72.11	92.18	69.8	90.98	76.57	93.84
MSHy-Addition	61.88	88.33	71.12	92.6	71.12	**92.92**	72.44	**94.02**	**77.89**	92.34	**78.38**	**94.55**
MSHy-Convolution	**66.5**	**91.25**	69.97	92.43	**72.77**	92.64	**75.25**	93.17	70.96	**94.02**	75.58	94.02
B: Results on Extremely Low Resource Scenario												
Model	*LSTM*		*GRU*		*Stack-LSTM*		*Stack-GRU*		*BiLSTM*		*BiGRU*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	33.66	79.60	32.51	80.66	37.29	82.50	32.40	79.38	37.29	82.50	39.44	82.61
Grapheme	**55.94**	87.73	57.26	88.90	48.68	85.63	58.75	88.98	50.33	85.33	61.39	90.28
MSHy-Basic	43.40	83.61	59.74	89.70	61.39	89.75	61.22	90.04	60.40	89.54	62.38	90.81
MSHy-Concatenation	36.63	78.27	56.44	89.01	53.96	86.79	60.23	89.72	50.17	86.32	61.72	89.97
MSHy-Addition	52.64	86.62	60.56	90.01	**65.02**	**90.75**	62.87	90.41	**61.72**	**89.97**	63.53	90.17
MSHy-Convolution	55.12	**87.76**	**61.06**	**89.83**	57.92	88.91	**66.83**	**91.58**	61.39	89.76	**67.33**	**90.95**

transformer-based models as a base model to adapt our proposed approach.

Another critical observation is that the phoneme representation presented in the study [189] is ineffective for transliterating English loanwords and named-entities. This is because the adapted CMU pronunciation dictionary is trained to capture the corresponding target Manipuri accent, which is different from our objective to capture the phonetic aspect of English named-entities and loanwords pronunciation. As a result, we instead use the English (G2P) conversion toolkit, presented in Section 4.6.4 for generating phoneme representation for other subsequent experiments.

4.7.2 Main Results

Sub-tables *A* and *B* of the table 4.4 shows the performances of different RNN-based transliteration models on two different English-Manipuri resource scenarios:

Moderately Low and *Extremely Low* respectively. The first two rows on each sub-tables A and B show the performances of the pivoted (phoneme-based) and direct (grapheme-based) models. It is evident from the results that grapheme-based transliteration setup outperforms its phoneme-based counterpart for all the RNN-based models. A similar observation is also reported in the earlier study [123] for the language pair. In the moderately low resource setting, RNN-based grapheme models achieve an improvement over their phoneme counters ranging from 23.60% to 27.89% in word accuracy, 7.37% to 8.70% in character accuracy. Similarly, each grapheme-based model significantly outperforms respective phoneme-based models in both the CA and WA for the extremely low resource scenario.

Further, the proposed multi-source RNN-based transliteration models with basic aggregator [252] (MSHy-Basic) outperforms the phoneme and grapheme models for all the cases on both the resource scenarios, except in only two instances while using the LSTM encoder:

1. In the moderately low resource scenario, CA of MSHy-Basic underperforms grapheme-model.

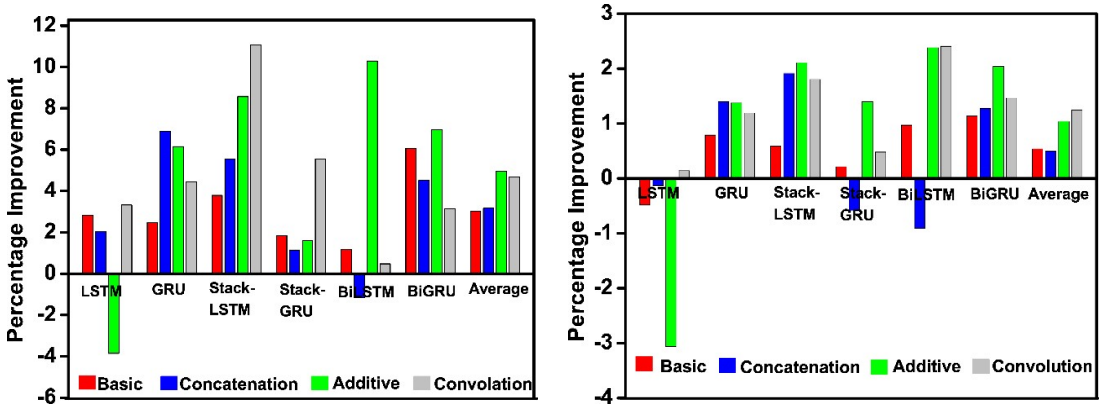
2. In the case of the extremely low resource scenario, grapheme-model outperforms MSHy-Basic in both the WA and CA.

This results show the efficacy of the proposed hybrid RNN-based transliteration models in taking advantages of multiple source sequences.

Having observed positive responses from MSHy-Basic, we further proposed another three aggregation methods (discussed in section 4.5.1), namely MSHy-Concatenation, MSHy-Addition, and MSHy-Convolution. For both the resource scenarios, it is also observed that the proposed aggregators outperform their

phoneme counterparts in all the cases and grapheme counterparts in the majority of the cases. Specifically, as shown in Figure 4.7, the MSHy-Concatenation is the only aggregation method that fails to surpass the MSHy-Basic in majority of the cases as compared to grapheme-models. The MSHy-Concatenation exceeds MSHy-Basic in only 50% of the cases in the moderately low resource scenario, while the MSHy-Basic dominates MSHy-Concatenation for all the cases in the extremely low resource scenario. The results show that the relatively straightforward aggregation method, i.e., MSHy-Concatenation, fails to handle the difference in importance of the phoneme and grapheme representations. On the contrary, other advanced aggregations methods (MSHy-Addition and MSHy-Convolution) dominate MSHy-Basic in 67% and 83.3% of the cases for the moderately low resource scenario. Similarly, in the case of the extremely low resource scenario also, MSHy-Addition and MSHy-Convolution dominate the MSHy-Basic in the majority of the cases. Overall, at least one of the aggregators outperforms MSHy-Basic. These results empirically provide insight into the importance of the aggregator. It also illustrates the effectiveness of the proposed aggregation methods. The best performance is obtained with MSHy-Addition using a BiGRU encoder in the case of the moderately low resource scenario. While, in the case of the extremely low resource setting, the MSHy-Convolution using BiGRU encoder secured the best result.

Figure 4.7 and figure 4.8 further show the percentage improvement of different RNN-based hybrid models over grapheme model and phoneme model, respectively, on the original moderately low resource scenario. From these figures, it is also evident that the proposed RNN-based hybrid models achieved an improvement as high as 11.08% in word accuracy and 2.38% in character accuracy over



(a) Word Accuracy (b) Character Accuracy

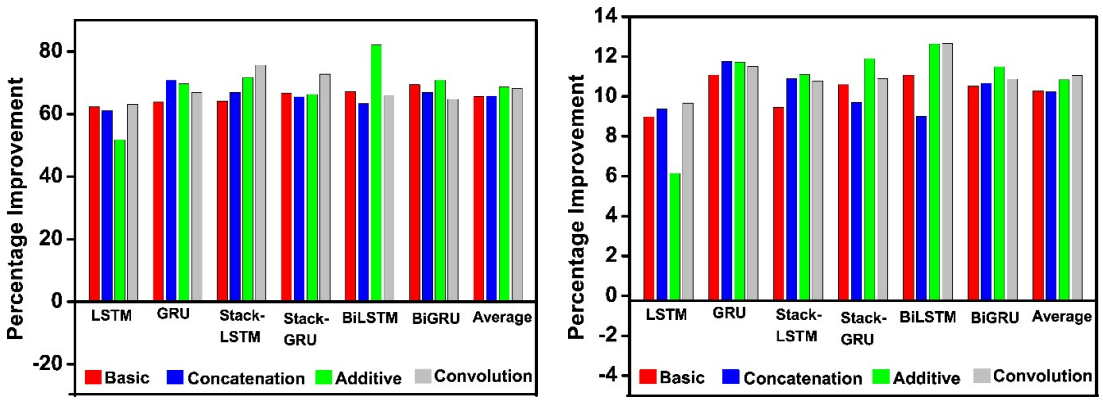
Figure 4.7: Performance improvement of different proposed RNN-based hybrid models over their grapheme-based counterpart on Moderately Low English-Manipuri resource scenario.

grapheme model, and 82.24% in word accuracy and 12.65% in character accuracy over phoneme model. The *Average* bars represent the average percentage improvement across different encoders for each aggregation method. On average, MSHy-Addition dominates others in word accuracy, and MSHy-Convolution dominates others in character accuracy.

Table 4.5: Performance of different Transformer-based Models on English-Manipuri language pair.

A: Results on Moderately Low Training Data												
Size	***Model Dimension = 256***						***Model Dimension = 512***					
Model	*2-Layer*		*4-Layer*		*6-Layer*		*2-Layer*		*4-Layer*		*6-Layer*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	37.79	81.31	39.60	82.86	23.27	71.82	36.47	80.04	28.22	75.85	21.12	68.33
Grapheme	66.01	91.17	69.64	92.53	35.31	75.74	60.23	88.78	46.04	83.48	26.24	71.40
MSHy-Serial	**69.31**	**91.74**	**71.45**	**92.75**	**38.45**	**77.14**	**60.89**	**89.53**	**53.30**	**86.85**	**30.36**	**74.65**
B: Results on Extremely Low Training Data												
Size	***Model Dimension = 256***						***Model Dimension = 512***					
Model	*2-Layer*		*4-Layer*		*6-Layer*		*2-Layer*		*4-Layer*		*6-Layer*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	28.88	78.12	75.67	27.56	13.86	59.70	24.59	75.77	24.09	73.54	10.89	58.39
Grapheme	52.97	86.34	46.04	83.82	15.84	60.86	43.89	83.24	40.59	80.63	16.17	63.59
MSHy-Serial	**55.94**	**87.25**	**46.20**	**84.74**	**18.81**	**63.13**	**45.21**	**83.19**	**41.58**	**82.11**	**16.83**	**63.07**

Table 4.5 presents different transformer-based models' performances on the two resource scenarios. Here also, the first two rows of the sub-tables (A and

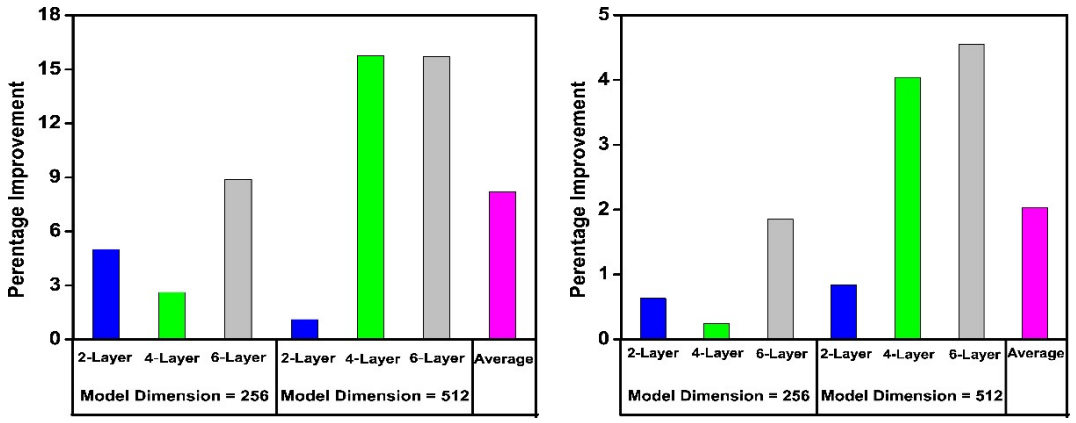


(a) Word Accuracy (b) Character Accuracy

Figure 4.8: Performance improvement of different proposed RNN-based hybrid models over their phoneme-based counterpart on Moderately Low English-Manipuri resource scenario.

B) show the performances of the phoneme-based and grapheme-based models. Akin to the observations achieved by RNN-based models, an improvement ranging from 24.22% to 75.83% in word accuracy and from 4.49% to 12.13% in character accuracy is also achieved by transformer-based grapheme models over the corresponding phoneme counterparts in case of the moderately low setting. Similar improvements are also observed for grapheme model over the phoneme model respective on the extremely low resource scenario.

Further, figure 4.9 and figure 4.10 show the percentage improvement of different transformer-based hybrid models over grapheme model and phoneme model respectively on the moderately low resource setting. The proposed multi-source transformer-based transliteration model (MSHy-Serial) also outperforms the phoneme and grapheme models for all the model settings. MSHy-Serial achieves an average increase of 71.45% in WA and 11.30% in CA over the phoneme models and an average increase of 8.18% in WA and 2.03% in CA over the grapheme counterparts. Similar improvements are also achieved on the extremely low resource scenario for the transformer-based hybrid models over the corresponding grapheme-based

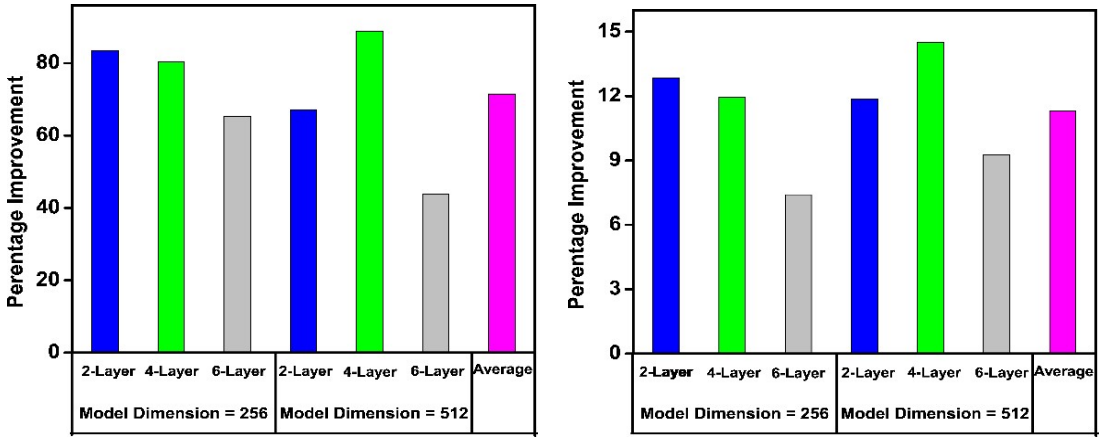


(a) Word Accuracy (b) Character Accuracy

Figure 4.9: Performance improvement of different proposed transformer-based hybrid models over their grapheme-based counterpart on Moderately Low English-Manipuri resource scenario.

model (on average 5.71% in WA and 1.14% in CA) and phoneme-based model (on average 68.02% in WA and 9.31% in CA), as shown in table 4.5. It is also observed that the transformer models with the model dimension of 256 outperform the 512 dimension settings. The models with six-layer underperform the two-layer and four-layer settings. MSHy-Serial with model dimension 256 and four-layer architecture provides the best performance obtaining 69.31% in WA and 71.45% in CA among the transformer-based models on the moderately low resource scenario. In comparison, MSHy-Serial with model dimension 256 and two-layer architecture achieved the best performance among the transformer-based models in the extremely low resource scenario.

It is evident from the above observations that the proposed hybrid multi-source encoder-decoder models can effectively combine characteristics of both source grapheme sequence and source phoneme sequence for both RNN and transformer-based models. If we compare RNN-based models with transformer-based models, we found that most RNN-based models dominate the transformer-based models.



(a) Word Accuracy (b) Character Accuracy

Figure 4.10: Performance improvement of different proposed transformer-based hybrid models over their phoneme-based counterpart on Moderately Low English-Manipuri resource scenario.

The best performing transformer-based model (MSHy-Serial with 4-layer architecture and model dimension of 256) surpasses only 50% (in terms of both the WA and CA) of the RNN-based hybrid models on the same resource scenario (moderately low). On the other hand, the best performing RNN-based model outperforms all transformer-based models on the same resource scenario. Transformer-based models have achieved breakthrough results in various NLP tasks outperforming RNN-based models. However, the outcome is different when only a limited training corpus is available. Similar to our problem, authors in [59] have also shown that the RNN-based models outperform transformer-based models on intent classification task when trained on a limited training corpus. Similarly, [230] have also demonstrated that transformer models perform better than RNN-based models on Historical Spelling Normalization only when provided with more training data. Even if our experiments suggest that RNN-based models are better than transformer-based models for the transliteration task, it is still early to conclude anything as the transformer-based models' poor performance compared to RNN-

based models may be due to hyperparameter choice [231]. A thorough investigation is necessary before drawing any conclusion in this respect, which is beyond the scope of this study. Nevertheless, we intend to answer this question in our future research.

Table 4.6: Transliterations predicted by the GRU based bi-directional framework with ground truth. Red color underline character shows the misclassified one.

Source	Source Phoneme	Target	Phoneme	Grapheme	Basic	Concatenation	Additive	Convolution
handicrafts	HH AE N D IY K R AE F T S	হেণ্ডিক্রাফ্টস	হন্দিক্রাফ্ট	হেণ্ডিক্রাফ্‌স	হেণ্ডিক্রাফ্‌স	হেণ্ডিক্রাফ্টস	হন্দিক্রাফ্টস	হান্দিক্রাফ্টস
carlsberg	K AA R L Z B ER G	কার্লসবর্গ	কার্লসবর্গ	কার্‌সবর্গ	কার্লসব্েের	কার্লসবর্গ	কার্লসবর্গ	কার্লসবর্গ
reis	R IY Z	রীস	রীস	র্েস	রীস	রীস	রীস	রীস
statement	S T EY T M AH N T	ষ্টেটমেন্ট	ষ্টেটমেন্ট	ষ্ট্েটমেন্ট	ষ্টেটমেন্ট	ষ্টেটমেন্ট	ষ্টেট্েমেন্ট	ষ্টেট্েমেন্ট
trapezoid	T R AE P AH Z OY D	ত্রাপেজোইদ	ত্রাপােসাইদ	ত্র্েপেজোদ	ত্রাপেজোইদ	ত্রাপেজোইদ	ত্রাপেজোইদ	ত্রাপেজোদ

To illustrate the transliteration pattern, table 4.6 shows outputs of different transliteration models using BiGRU neural encoder of a few unseen named entity words and loan words. The results show that the grapheme-based model fails to transliterate the target grapheme correctly because of its inability to capture some of the phoneme information. For instance, the named-entity *reis* (রীস) is wrongly transliterated as রেস because it fails to capture characteristics of *ei* graphemes combination. However, multi-source models are able to resolve such issues by taking advantage of the phoneme representation (/IY/). Similar characteristics are also seen for other words, as shown in table 4.6.

4.7.3 Evaluation on Other Language Pairs

To evaluate whether the improvement of the proposed hybrid model on resource-poor English-Manipuri language pair could also be achieved for other language pairs with a relatively larger corpus. We further investigate the performance of the proposed model on four other distinct language pairs transliteration task, namely, *English-to-Chinese* (En-Ch), *English-to-Thai* (En-Th), *English-to-Persian* (En-

Table 4.7: English-Chinese (En-Ch), English-Chinese (En-Ch), and English-Persian (En-Pe) Language Pairs Dataset Description

	En-Ch	*En-Th*	*En-Pe*	*En-Hi*
Training	42218	30529	17570	8997
Testing	998	927	1000	690

Table 4.8: Performances of RNN-based Models on other language pairs

Model	*En-Ch*		*En-Th*		*En-Pe*		*En-Hi*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	45.04	75.57	38.30	80.87	41.8	84.39	24.20	74.55
Grapheme	68.28	86.21	37.04	80.28	55.5	89.29	39.78	82.95
MSHy-Basic	67.74	85.90	46.02	85.48	57.50	89.77	**42.90**	83.98
MSHy-Concatenation	68.74	**86.75**	47.03	85.04	55.00	89.31	40.43	**84.09**
MSHy-Addition	**69.01**	86.28	**49.30**	**85.50**	56.70	89.42	39.28	83.29
MSHy-Convolution	67.64	86.18	44.01	84.88	**57.60**	**89.96**	39.69	83.00

Pe), and *English-to-Hindi* (En-Hi). We use the publicly available dataset provided by the NEWS 2018* shared task for all these language pairs. These language pairs are explicitly chosen as they provide four distinct resource scenarios, which are relatively larger than the English-Manipuri language pair. Moreover, it facilitates using the same source language English G2P toolkit (used in this study), and be consistent with our English-Manipuri experiments. A complete description of the dataset is presented in the study [30]. Since the G2P toolkit does not support multi-words, we remove all the multi-word pairs from the dataset. The official development set in NEWS 2018 is used as the test set, and development sets are created by randomly selecting 20% of the training data. A detailed description of the dataset used in this study is presented in Table 4.7.

We choose the best performing RNN-based model *(i.e., BiGRU)* to investigate the performance of the proposed hybrid transliteration model on these language pairs. The model configurations are kept the same as the one described in Sec-

*http://workshop.colips.org/news2018/

Table 4.9: Performance of different Transformer-based Models on other language pairs.

Language	***En-Ch***						***En-Th***					
Model	*2-Layer*		*4-Layer*		*6-Layer*		*2-Layer*		*4-Layer*		*6-Layer*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	52.91	78.67	46.74	75.24	27.35	59.5	30.53	76.70	26.75	**74.26**	**14.56**	**62.07**
Grapheme	64.83	84.39	61.52	82.96	37.07	68.19	30.64	75.07	25.46	71.19	10.14	55.88
MSHy-Serial	**66.13**	**85.18**	**62.12**	**83.42**	**43.69**	**71.46**	**34.09**	**78.24**	**27.29**	72.14	11.97	58.65

Language	***En-Pe***						***En-Hi***					
Model	*2-Layer*		*4-Layer*		*6-Layer*		*2-Layer*		*4-Layer*		*6-Layer*	
	WA	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**	**WA**	**CA**
Phoneme	36.2	82.00	38.00	82.76	18.50	68.67	17.22	69.22	14.76	69.41	6.51	55.15
Grapheme	46.78	86.58	43.30	84.41	**24.10**	**73.26**	30.53	78.74	24.75	75.93	9.26	60.25
MSHy-Serial	**47.35**	**86.58**	**46.40**	**86.50**	20.90	69.67	**30.53**	**78.77**	**28.51**	**78.50**	**11.72**	**60.57**

tion 4.6.3. However, considering the increase in the size of the training corpus, we increase the decoder's hidden size to 1024, and the embedding dimension to 512. In the transformer-based model, the models are tested on all three different layer settings (2-layer, 4-layer, and 6-layer). Similar to the RNN-based model, we kept all the hyper-parameters values the same as discussed in Section 4.6.3, except for the model dimension and feed-forward dimension, which are fixed to 512 and 1024 respectively.

Table 4.8 and table 4.9 show the performance of the RNN-based and transformer-based transliteration models on the language pairs. Similar to the results obtained with the English-Manipuri language pair, the proposed hybrid models consistently outperform both the grapheme and phoneme counterparts, except for a few cases. However, it is observed that for all the language pairs, the best-performing model turns out to be one of the proposed hybrid configurations surpassing both the respective grapheme-based and phoneme-based models. This empirically shows the proposed hybrid transliteration model is also applicable for other language pairs. Interestingly, the phoneme-based model outperforms the grapheme-model for the *En-Th* language pair for the majority of the model settings. In this case, all the

proposed RNN-based hybrid model outperforms the RNN-based phoneme-model, improving up to 28.72% in WA and 5.72% in CA. Similarly, on the transformer side, the best hybrid model (MSHy-Serial with two-layer setting) beats the best single-source model (phoneme-model with two-layer setting) in both the WA and CA.

4.8 Summary

This chapter proposes a neural hybrid multi-source encoder-decoder transliteration model suitable for integrating the phoneme sequence and the grapheme sequence. We investigate the proposed model on both the RNN-based and transformer-based encoder-decoder frameworks. We further propose various aggregation methods that better combine the incoming information obtained from multiple sources in RNN-based models. The proposed transliteration models are then compared with their phoneme and grapheme based counterparts. The proposed models' performance is investigated over a resource-poor English-Manipuri language pair for transliterating named-entities and loanwords. From various experimental setups, it is evident that the multi-source encoder-decoder transliteration model can effectively integrate the characteristics of both phoneme sequence and grapheme sequence, and it outperforms its phoneme and grapheme based counterparts. We further investigate the effectiveness of the proposed model on four other language pairs (English-Chinese, English-Thai, English-Persian, and English-Hindi) to determine its ability to generalize. For all the language pairs as well, the proposed hybrid models outperform their baseline phoneme and grapheme models.

5

Empirical Study of Unsupervised Cross-lingual Embedding Methods

This chapter presents an extensive evaluation of two popular unsupervised approaches of inducing cross-lingual word embeddings, namely MUSE and Vecmap, on the comparable corpus presented in Chapter 3. The study in this chapter is primarily motivated by two reasons:

1. Despite using cross-lingual embeddings in various NLP tasks, including un-

supervised machine translation, there is no related study for the Manipuri-English language pair in the literature.

2. To determine whether our generated corpus presented in Chapter 3 is feasible for generating a robust cross-lingual embeddings between the language pair.

From various experimental results, it is observed that the proposed corpus derived from news publications can be used to build effective cross-lingual embeddings between Manipuri and English. The results also show that the Vecmap consistently outperforms the MUSE. In addition, this study also presents methods to enhance the embeddings further. A Manipuri suffix segmenter is proposed to segment words into the root and suffixes. The proposed segmenter can alleviate the Manipuri language's morphological inflection problem. Instead of a Manipuri-English dictionary, a novel method is also proposed to utilize automatically generated transliterated word pairs to further enhance the embeddings.

5.1 Introduction

The representation of words in cross-lingual vector spaces, called cross-lingual word embeddings (CLWEs) [148] is becoming increasingly popular. CLWEs allow us to compare word meanings across languages. Moreover, by providing a common representation space, they facilitate cross-lingual models transfer between languages, mainly from rich-resource languages to low-resource languages. Subsequently, CLWEs have been used in several downstream tasks like cross-lingual information retrieval [242], cross-lingual text classification [112], bilingual dictionary induction [43], etc. In fact, they form the basis of the UMT models [43, 14, 12]

on which this thesis is based on.

Over the years, several approaches have been proposed for inducing CLWEs, each requiring different forms of cross-lingual supervision [186]. Unsupervised CLWEs models are exciting as they rely only on inexpensive source and target language non-parallel texts to learn the embeddings. However, a systematic comparison of these models is missing for the Manipuri-English language pair. Moreover, a sizeable monolingual corpus for Manipuri is currently not available. We rely on a modest size unexplored Manipuri-English comparable corpus presented in Chapter 3 obtained from news publications to generate the CLWEs. Considering the above reasons, investigating the unsupervised models on the language pair is necessary. This study fills this void by empirically comparing two popular unsupervised cross-lingual word embedding models: (1) *MUSE* (Multilingual Unsupervised and Supervised Embeddings) [43] and (2) *Vecmap* [11], on bilingual dictionary induction (BDI) task. Preliminary investigations confirm that the unsupervised methods can generate an effective CLWEs using the proposed comparable corpus, showing that the proposed corpus is feasible for cross-lingual studies between the language pair. We also observe that the Vecmap model performs consistently better than the MUSE on the language pair.

On top of analyzing the performance of previous models, we further enhance the embeddings. This study proposes a Manipuri Suffix Segmenter that normalized the agglutinative nature of Manipuri by segmenting words into root and suffixes. The proposed segmenter is developed by enhancing the language-independent stemmer, the GRAph-based Stemmer (GRAS). As the bilingual dictionary required for generating cross-lingual embeddings between Manipuri and English is not readily available, this study also presents a method to deploy au-

tomatically generated transliterated word pairs using transliteration models to further enhance cross-lingual embeddings. From various experiments, it is observed that the proposed techniques significantly outperform the corresponding baselines.

5.2 Related Studies

5.2.1 Cross-lingual Word Embeddings

Word embeddings have proven to be one of the most widely used resources in NLP for modeling linguistic phenomena in both supervised and unsupervised settings [149]. Similarly, word embeddings counterpart in multiple language settings, the CLWEs, have also been extensively used and studied in recent years. Previous approaches to obtaining CLWEs vary with respect to the use of supervision signals [186]. Earlier studies depend on parallel-data supervisions like sentence-aligned corpus [133], document-level alignments [84], etc. Few uses expensive lexical resources like WordNet, ConceptNet, etc. [225, 150], while some require both sentence and word alignments [139]. However, such resources are currently not available for Manipuri.

For learning CLWEs, a recently developed branch of research uses independently trained monolingual source and target languages word embeddings. These embeddings are then mapped to a shared space. Existing mapping-based approaches include [10, 186]:

1. *Regression methods* map the embeddings in one language to another language [148]. The mapping is accomplished using an objective function that learns the linear transformation minimizing the sum of squared Euclidean

distances for the bilingual dictionary entries. Other researchers have enhanced the model by incorporating L2 regularization [51, 241].

2. *Canonical methods* map the source and target language embeddings to a shared space from both directions where their similarity is maximized. This is usually done through Canonical Correlation Analysis (CCA). Authors in [136] build on this work by applying Deep CCA to the learning of non-linear mappings. The method was also extended to the multilingual by considering the pivoted approach [7].

3. *Orthogonal methods* constrain the transformation matrices that are used for aligning the embeddings to be orthogonal.

4. *Margin methods* [51], as a way to address the hubness problem, map the embeddings in one language to maximize the margin between the correct translations and the rest of the candidates.

Mapping-based approaches require source and target language monolingual embeddings and a dictionary, if any at all, to learn high-quality cross-lingual embeddings. MUSE [43] and Vecmap [11] are the state-of-the-art unsupervised mapping-based approaches that can generate CLWEs without using any parallel resources [186]. These models have paved the way for the creation of unsupervised machine translation systems that do not rely on parallel corpora [14, 13]. This chapter provides a systematic comparison of MUSE and Vecmap on the low-resource Manipuri-English language pair.

5.2.2 Evaluation of Comparable Corpus Quality

The quality of comparable corpora can be evaluated by estimating a cross-lingual similarity between the bilingual texts. Studies in [221] have proposed several approaches to measure the comparability between Wikipedia articles written in different languages. Similarly, authors in [90] have proposed a cross-lingual information retrieval based similarity measure to construct a strongly comparable news corpus. However, such evaluation schemes rely on a similarity score based on the number of translation equivalents between the bilingual texts. The translation equivalents are obtained from a sizeable bilingual resource either in the form of parallel sentences or bilingual dictionaries that are known to be identical in terms of topic, thematic, genre, time, meaning, etc. Manipuri-English being an extremely low-resource language pair, such bilingual resources are not available presently.

Over the years, comparable corpora have been utilized as a secondary resource to enhance the BDI and MT models trained with limited parallel resources. As reported in [180], the surge in using comparable corpora as the primary resource started with the advent of CLWEs [149, 241]. The main advantage of CLWEs is that they can be learned with little or no parallel bilingual data by utilizing only comparable corpora, making them perfect for determining the corpus quality. We investigate the usability of the proposed Manipuri-English Comparable Corpus by analyzing the performances of the standard BDI models based on CLWEs (MUSE and Vecmap). The choice of BDI for evaluating the corpus is also motivated by the previous study [226] that is, the BDI scores correlate with human judgments of cross-lingual similarity. Detailed description of the models are presented below.

5.3 Unsupervised CLWE Methods

5.3.1 MUSE

MUSE [43] is a mapping-based method where the objective is to learn a mapping between independently trained monolingual source language word embeddings X and target language word embeddings Z to generate the shared embedding space or CLWEs. The goal is to find a transformation matrix W that minimizes the following Euclidean distance over a dictionary D:

$$W = \arg\min_{W} \sum_{i,j \in D} ||X_i W - Z_j||_F \tag{5.1}$$

MUSE initializes a seed dictionary D solely from monolingual data using Generative Adversarial Networks (GANs) [65]. In this method, a discriminator is trained to discriminate samples from the mapped source embeddings WX from the target embeddings Z, while W is simultaneously trained to prevent this. The estimated W is then used to build a small bilingual dictionary. The entire process undergoes iterative Procrustes* refinement using the new transformation matrix to create a new dictionary until convergence. MUSE depends heavily on the assumption of approximate isomorphism between the source and target language embeddings, which frequently leads to poor GAN-based initialization, particularly for distant languages [63].

*https://en.wikipedia.org/wiki/Orthogonal_Procrustes_problem

5.3.2 Vecmap

Vecmap [11] follows similar concept to MUSE. The method first induces an initial dictionary (D) based on the assumption that the monolingual vector spaces will be isometric [149]. D is obtained in an unsupervised manner by exploiting the intra-lingual similarity distribution of individually trained source language word embeddings X and target language word embeddings Z. Using the initial seed dictionary (D), the orthogonal transformation matrices W_X and W_Z are learned to map X and Z into a shared embedding space. The objective is to minimize the following function:

$$\hat{W}_X, \hat{W}_Z = \arg \min_{W_X, W_Z} \sum_{i,j \in D} ||X_i W_X - Z_j W_Z||_F \tag{5.2}$$

Similar to MUSE, this training process is iteratively refined by using the estimated matrices ($\hat{W}_X$, $\hat{W}_Z$) to create a new seed dictionary (D). D is generated by using Cross-domain Similarity Local Scaling (CSLS) [43]. For each word pairs (X_i, Z_j), we update $D_{i,j} = 1$ if the CSLS score between them is the highest over all combinations of X_i and other target words. Otherwise, $D_{i,j} = 0$. The dictionary is induced for both the directions, and then concatenated together [11]. Vecmap adopts multi-step pre-processing (unit length normalization, mean centering, and ZCA whitening) and post-processing steps (cross-correlational re-weighting, de-whitening, and dimensionality reduction) as in the study [10]. Moreover, the model employs stochastic dictionary induction where elements in D are randomly set to 0, allowing the model to escape poor local optima.

Algorithm 1 Manipuri Suffix Segmenter

1: Identify suffix pairs (s_1, s_2) as a candidate pair if

- both the s_1 and s_2 satisfy the Manipuri suffix constraints.
- there are word pairs of the form ($w_1 = ps_1$, $w_2 = ps_2$) that share a sufficiently long prefix p.
- there are sufficient number of other word pairs of the form ($w_a = p's_1$, $w_b = p's_2$) having a common prefix p' followed by the suffixes (s_1, s_2).

2: Tag word pairs as morphologically related if

- they share a non-empty common prefix.
- the suffix pair that remains after the removal of the common prefix must be a candidate pair.

3: Model word relationships in the form of a graph G, where the words represent nodes, and edges are the connection between the morphologically related word pairs.

4: **repeat** ▷ Obtain morphologically related words classes.

- Identify pivot word - node with maximum degree.
- Words that are connected to the pivot is put in the same class as the pivot if they shares many common neighbours with the pivot.
- Remove and group all the words in the class as a morphologically related word class.

until G is empty

5: Stem each words in a class by mapping it to the pivot.

6: **for** each class **do** ▷ Segmentation Module

7: **for** each word in the class other than the pivot **do**

∗ separate the root and its suffix by determining the longest common prefix between the words and the pivot, such that the suffix pair after removing the prefix is a valid suffix candidate pair

8: Segment the pivot by choosing the longest suffix associated with it among all the pivot-word suffix pairs present in the class

5.4 Manipuri Suffix Segmenter

In Manipuri, words are primarily associated with suffixes based on the number, gender, and other factors. There are no infixes, and suffixes are more frequent than prefixes [161]. Such inflections generate a large vocabulary, resulting in a large number of unseen and low-frequency words. This is a severe issue as unsupervised CLWEs models depend on frequency-based co-occurrence features. To alleviate the issue, we present a Manipuri suffix segmenter that segments words into the root and their suffixes.

Few earlier studies have reported the development of Manipuri stemmers [161, 145] and morphological analyzers [40, 211] using expensive handcrafted rules.

However, these systems are not available for public use. Considering the unavailability of Manipuri word segmenters, we decide to propose a Manipuri word suffix segmenter by modifying GRAph-based Stemmer (GRAS), a popular language-independent stemmer [166]. A detailed description of the proposed segmenter is presented in Algorithm 1. GRAS clusters words into a group of morphologically related classes based on word prefixes and suffixes. However, we observe that some of the candidate suffixes identified by GRAS are linguistically invalid. For example, we found that invalid sub-words like ◌া, ◌ুগী, দ, etc. are also identified as valid suffixes. To overcome this issue, we incorporate two linguistically motivated inexpensive Manipuri suffix constraints: (1) suffix's length must always be greater than one, and (2) suffix must not begin with a vowel. These constraints are imposed while generating the candidate pairs.

After generating morphologically related classes, each word in a class is stemmed by mapping them to the pivot of that class in the original GRAS. However, we segment each word instead of stemming by separating the root and its suffix. Precisely, we determine the longest common prefix between the words and the pivot, such that the suffix pair after removing the prefix is a valid suffix *candidate pair* (refer Algorithm 1). The pivot word is also segmented by choosing the longest suffix associated with it among all the pivot-word suffix candidate pairs present in the class—this procedure is iterated for all the morphological-related classes.

5.5 Improving Cross-lingual Word Embeddings using Transliteration Word Pairs

Cognates like proper names and digits have been extensively exploited to enhance the CLWEs [186]. Similarly, many cross-lingual embedding methods also utilized a bilingual dictionary between the source and target language pair [186]. However, as the proper names and the digits are also written in respective scripts (Manipuri and English orthographies), we cannot directly exploit the cognates. Moreover, as a bilingual dictionary between the language pair is unavailable, this study exploits automatically generated transliterated word pairs to enhance cross-lingual embeddings.

This study considered the Vecmap as the based model to incorporate the transliteration features as it performs superior to the MUSE in our preliminary investigations. We formulate the mapping between the source and target language embeddings in the Vecmap as a semi-supervised method by utilizing transliteration word pairs to populate the initial dictionary D. These transliteration pairs are generated automatically by using transliteration models (discussed in the previous chapter). If the character accuracy (CA) between a source language word and a transliterated version of a target language word towards source language is greater than a certain threshold (T_θ), then the source and target word pairs can be classified as transliteration word pairs [235]. In this study, we consider both the Manipuri-to-English and English-to-Manipuri transliteration models to identify the transliteration word pairs, Specifically, a cell in the initial dictionary $D_{i,j} = 1$ if $\text{CA}(x'_i, z_j) \geq T_\theta$ and $\text{CA}(x_i, z'_j) \geq T_\theta$, where x_i and z_j. are source and target words, respectively x'_i and z'_j represents the transliterated versions of x_i and z_j respectively

Table 5.1: Manipuri-English News Domain Comparable Corpus.

Language	Documents	Sentences	Words	Vocabulary	Segmented Vocabulary
English	13408	136560	5.79M	80855	80855
Manipuri	13117	273108	5.62M	277406	165998

towards the other language using the transliteration models. Otherwise, we set D_{ij} equal to 0. For all the experiments, we consider $T_\theta = 100\%$.

5.6 Experimental Setup

5.6.1 Dataset Description

All the models in this study are trained using our proposed comparable Manipuri-English corpus presented in Chapter 3. Table 5.1 shows a detailed description of the corpus. The lower-cased English texts are tokenize using Moses Tokenizer*, while a simple white-space tokenization scheme† is used for Manipuri texts. Manipuri text in the corpus are segmented by applying the proposed suffix segmenter. Table 5.1 sixth row (Segmented Vocabulary) shows the vocabulary size of the segmented corpus.

The availability of a reliable evaluation benchmark is necessary for the rapid progress of any NLP task. However, the Manipuri-English language pair does not have any reliable evaluation dataset for the BDI task to the best of our knowledge. In this work, we also introduce a BDI test dataset for the news domain to systematically track its progress. The test set consists of 981 pairs of words manually created by native speakers. As many source language words have several translations on the target language side, the actual number of unique source

*https://github.com/moses-smt/mosesdecoder
†Punctuation symbols are first separated.

language words is smaller than the total number of translation pairs. In our BDI test set, the number of unique English words is 858, while there are 856 unique Manipuri words. The evaluation dataset words were chosen such that it consists of a mixture of classes (part-of-speech) to properly evaluate models' performance. The dataset's percentage of common nouns, proper nouns, verbs, adjectives, and adverbs is 67.6, 4.55, 10.4, 9.4, and 3.72, respectively. Preposition, pronoun, and number together constitute the remaining. The distribution is similar to other BDI test data for other languages [43, 105]. Similar to other previous studies [43, 11] on the BDI task, we have also not considered multi-words as the standard CLWEs operate on word-level. However, single word named entities and terminological units is present.

5.6.2 Transliteration Model

We consider the bi-directional GRU encoder-decoder grapheme-based transliteration setup discussed in the previous chapter. We fixed the hidden layer's size and the embedding dimension to 512 and 256, respectively. The models are trained using the dataset presented in our transliteration study (refer Table 4.2). It consists of 4428 training transliteration pairs, 1000 development pairs, and 607 testing pairs. The models gives a word accuracy of 73.27% for English-to-Manipuri transliteration and a word accuracy of 61.7% for Manipuri-to-English transliteration on the testing dataset.

5.6.3 Models Configurations

We use the original distributions of both the MUSE* and the Vecmap†. For all the experiments, the source and target language embeddings are obtained using the fastText‡ [24]. The dimension of the embeddings is fixed to 300. Other hyperparameters values are kept the same as the original settings. Given a source language word, its corresponding translation is retrieved through (1) *Cross-domain Similarity Local Scaling (CSLS)* [43], and (2) *Nearest Neighbour (NN)*, search of target language words. The models are evaluated by using the precision at k *(P@k)* evaluation metric. The models' performance is evaluated by using the precision at k *(P@k)* and *MAP* metrices. The P@k evaluates the percentage of correct test pairs among the k highest ranked candidates. We report *P@k* for k = 1 and 5 in percentage.

5.7 Results and Discussion

5.7.1 Suffix Segmenter Evaluation

We first intrinsically evaluate the proposed suffix segmenter. The evaluation is performed by using a manually segmented randomly chosen 200 unique Manipuri words. It is observed that the suffix segmenter obtains an accuracy of 77%. Moreover, the segmented Manipuri vocabulary size decreases by 40.16% as compare to the original (refer Table 5.1). It also indirectly indicate the effectiveness of the proposed Manipuri suffix segmenter. Some segmentation examples of morphological

*https://github.com/facebookresearch/MUSE
†https://github.com/artetxem/vecmap
‡https://fasttext.cc/

Table 5.2: Some Segmentation examples.

Noun		Verb	
Non-segmented	**Segmented**	**Non-segmented**	**Segmented**
বাজারসু	বাজার সু	পীরুসি	পীরু সি
বাজাররোমদা	বাজার রোমদা	পীরুরবসু	পীরু রবসু
বাজারনচিংবা	বাজার নচিংবা	পীরুদ্রবদি	পীরু দ্রবদি
বাজারদসু	বাজার দসু	পীরুরবদি	পীরু রবদি
ষ্টেটশিং	ষ্টেট শিং	ফোঙদোক্নবা	ফোঙদোক্ন বা
ষ্টেটতনা	ষ্টেট তনা	ফোঙদোক্নরকখি	ফোঙদোক্ন রকখি
ষ্টেটতসু	ষ্টেট তসু	ফোঙদোক্নরিবনি	ফোঙদোক্ন রিবনি
ষ্টেটনিনা	ষ্টেট নিনা	ফোঙদোক্নরে	ফোঙদোক্ন রে

Table 5.3: Ablation study to determine the impact of the linguistically motivated rules on the performance of the segmenter.

	Accuracy (%)
With both the constraints	77
Without constraint - 1	64
Without constraint - 2	61
Without both the constraints	57

variants of nouns (বাজার *, ষ্টেট†) and verbs (পীরু‡, ফোঙদোক্ন§) are shown in Table 5.2. The table also indicates that the effectiveness of the proposed segmenter in normalising the morphological inflection issue.

To evaluate the effectiveness of the two Manipuri suffix constraints (refer Section 5.4), we also perform an ablation study in which each constraint is removed individually to measure their impact on the performance of the segmenter. Table 5.3 shows the evaluation results. It is observed that the suffix segmenter with the constraints obtains an accuracy of 77%. However, if we do not consider the constraints, the accuracy decreases to 57%. This clearly shows the effectiveness

*Transliteration in Roman alphabet: bazar
†Transliteration in Roman alphabet: state
‡Transliteration in Roman alphabet: piru
§Transliteration in Roman alphabet: phongdokna

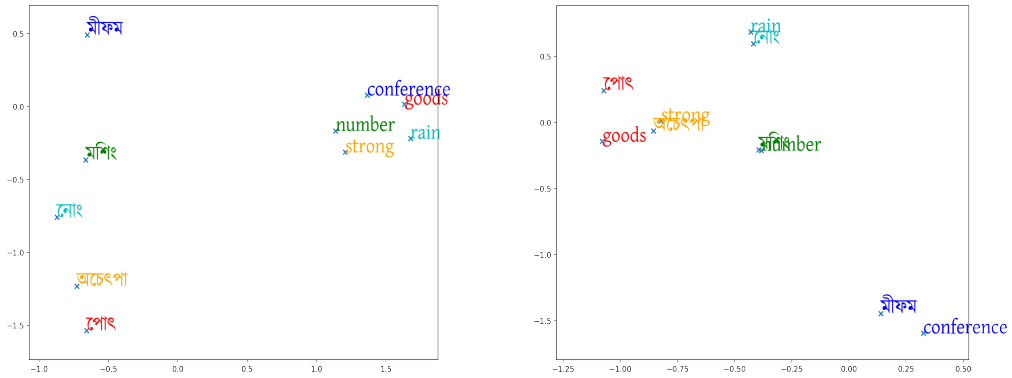
Table 5.4: Performances (Precision and MAP in percentage) of BDI models. CSLS stands for Cross-domain Similarity Local Scaling and NN for Nearest Neighbour.

	En → Mni					
	CSLS			*NN*		
	P@1	P@5	MAP	P@1	P@5	MAP
MUSE	22.70	36.87	29.47	20.37	31.39	25.93
+ Segmentation	31.02	46.87	38.40	28.79	42.42	35.15
Vecmap	25.14	41.42	32.78	24.68	37.92	31.20
+ Segmentation	33.10	50.90	41.34	32.27	48.54	40.00
+ Transliteration	**33.37**	**51.46**	**41.44**	**33.10**	**49.10**	**40.72**
	Mni → En					
	CSLS			*NN*		
	P@1	P@5	MAP	P@1	P@5	MAP
MUSE	31.62	45.32	38.39	28.00	41.35	34.38
+ Segmentation	37.07	52.56	44.59	31.12	47.03	38.61
Vecmap	36.99	48.71	42.72	34.07	47.08	40.63
+ Segmentation	40.80	56.13	48.37	36.65	53.77	45.00
+ Transliteration	**41.63**	**56.29**	**48.75**	**38.31**	**53.53**	**46.03**

of these constraints.

5.7.2 BDI Evaluation

Table 5.4 shows the performance of the CLWE models trained using the domain-aligned comparable corpus presented in Table 5.1 on the BDI task. We evaluate both the directions, namely English-to-Manipuri (*En → Mni*) and Manipuri-to-English (*Mni → En*). For each model, the rows denoted with *+Segmentation* in the table represent the respective BDI results on the segmented dataset. Similarly, the rows marked with *+Transliteration* show the performance of the transliteration word pairs incorporated version of the Vecmap on the segmented dataset. Upon comparing the MUSE and the Vecmap, it is observed that the Vecmap performs relatively better than the MUSE for the language pair. Similar results are also reported on the study [186] for other language pairs. We also observe that the



(a) Unaligned Embeddings **(b)** Aligned Embeddings

Figure 5.1: Visualisation of Manipuri-English CLWEs with PCA. Matching color words represent a translation pair.

CSLS retrieval method outperforms its neural network counterpart, the NN, for all the settings. Authors in [223] have shown that unsupervised systems fail for English-Spanish, English-Finnish, and English-Hungarian language pairs due to the difference in the domain between the source and target corpora. In our case, we achieve P@1 score of 25.14% for $En \rightarrow Mni$ direction and 36.99% for $Mni \rightarrow En$. Similarly, the Vecmap obtains a P@5 score of 41.42% for $En \rightarrow Mni$ and 48.71% for the $Mni \rightarrow En$ direction. This confirms that our corpus is feasible for generating effective cross-lingual embeddings. Moreover, we found that segmenting Manipuri text enhances the models' performance significantly for all the cases. The P@1 score for $En \rightarrow Mni$ increases on average by about 23% % compared to the non-segmented version. Similar increments are also observed for $Mni \rightarrow En$ directions. It is also evident from the table that semi-supervising the Vecmap further increases the results showing that exploiting automatically generated transliteration word pairs improves the CLWEs.

Figure 5.1 (a) displays the scatter plots of word embeddings of five transla-

Table 5.5: *Mni* → *En* BDI examples as given by the Vecmap (CSLS) with ground truth (References). Reference with multiple entries are separated by semicolon (;).

Word	**References**	**Top-1**	**Top-2**	**Top-3**	**Top-4**	**Top-5**
নোংপোক	east;eastern	northeast	northeastern	northeast	north	north-east
লাপ্পা	far	kilometres	metre	kilometre	bumpy	metres
অহল-লমন	elders	grandparents	fathers	guardians	parents	daughters

tion pairs (goods, পোৎ*), (conference, মীফম†), (number, মশিং‡), (rain, নোং§), and (strong, অচেৎপা) projected to two-dimension using PCA¶. Before alignment, the embedding of a word in English and embedding of the translated word in Manipuri do not possess any association between them. However, the Vecmap consistently brings the word and its translation closer on the shared embedding space, as shown in Figure 5.1 (b). The figure also clearly demonstrates the reliability of the proposed comparable corpus for the generation of CLWEs.

Upon manual examination, we also observe some interesting patterns which the model learns. Although the model fails to predict the correct target word, we found that the predicted words are mostly semantically similar words. A few such examples are shown in Table 5.5. If we consider such semantically similar words and the morphological variants as correct translations, then the P@1 for *En* → *Mni* and *Mni* → *En* of the Vecmap on the segmented corpus reaches up to 72.35% and 69.62% respectively, which is an excellent sign.

Although we can minimize morphological infections in Manipur, it remains a problem. Table 5.6 shows the top five predicted Manipuri words for the English words (road, time, after) given by the Vecmap on the segmented corpus with the

*Transliteration in Roman alphabet: pot
†Transliteration in Roman alphabet: mipham
‡Transliteration in Roman alphabet: masing
§Transliteration in Roman alphabet: nong
¶https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

Table 5.6: Top five predicted Manipuri words for the corresponding English words given by the Vecmap (CSLS) with ground truth.

Word	**Reference**	**Top-1**	**Top-2**	**Top-3**	**Top-4**	**Top-5**
road	লম্বী	লম্বীশিংদা	লম্বীশিং	লম্বী	লম্বীদা	রোদ
time	মতম	মতমদি	ঙাই	মতম	মতমদদা	মতমদ
after	মতুং	মতুংদা	মতুংদনি	মতুংদগী	মতুং	মতুংদনা

CSLS retrieval method. Even though the *top-1* translation and the *reference* mean are the same for all the English words, they have been wrongly classified because of the presence of morphological inflections. All the top five predicted words for the English word *after* are all related to the same root word মতুং. This also explains a huge difference in the performance between $En \rightarrow Mni$ and $Mni \rightarrow En$.

5.8 Summary

This chapter presents the first-ever attempt to generate CLWEs between the Manipuri-English language pair. We empirically investigate the performance of the popular unsupervised models (MUSE and Vecmap) on the language pair bilingual dictionary induction task. It is found that the Vecmap consistently outperforms the MUSE. We also show that a modest comparable corpus obtained from news publications can be used in place of a large source and target language monolingual corpus to generate robust CLWEs between the language pair. This study also proposes a Manipuri suffix segmenter that normalized the morphological inflection problem of Manipuri. A method is also proposed to exploit phonetically similar transliterated words to semi-supervised cross-lingual embeddings generation. The findings of our experiments suggest that the proposed methods improve both English-to-Manipuri and Manipuri-to-English bilingual dictionary induction.

6

Manipuri-English MT using a Comparable Corpus

This chapter presents a method for developing an MT system for Manipuri-English language pair without using parallel sentences. This study first empirically evaluates state-of-the-art unsupervised MT approaches on the language pair. Experimental results show that unsupervised statistical MT approach outperforms unsupervised neural MT approaches for the language pair. This chapter

further enhances the state-of-the-art unsupervised statistical MT model by incorporating:

1. Suffix segmenter to take care of the agglutinative nature of text in Manipuri language.

2. Manipuri-English machine transliteration to address the challenge of preparing a bilingual dictionary.

3. A novel method for exploiting a limited number of document-aligned comparable pairs.

The proposed method alleviates two of the core challenges in developing MT systems for low-resource languages; the challenges for preparing (i) sentence-level parallel corpus by using a comparable corpus and (ii) bilingual dictionary using transliteration models. From various experimental setups, it is evident that the proposed methods significantly outperforms their baseline counterparts.

6.1 Introduction

Building a MT system generally requires a large number of sentence-level parallel corpus (parallel sentences) [115, 16]. To avoid the problem of creating large volume of parallel corpus, researchers, in recent time, have started developing unsupervised machine translation (UMT) methods [14, 127]. UMT methods generally consider independently curated monolingual corpora for source and target languages without the need of sentence-level translated parallel sentences. Such models have shown to provide encouraging performance over rich resource languages like English, French, German, etc. [42, 224]. However, as observed in

[143, 108, 131], the effectiveness of such unsupervised methods may depend on linguistic similarity (like language branch, alphabet, morphology, etc.) between the source and target languages.

Considering the challenges in creating large volume of parallel corpus, unsupervised methods may be considered as a promising alternative for developing machine translation systems for low-resource languages. This chapter proposes a MT framework for Manipuri-English language pair by exploiting a comparable corpus. Motivated by recent advancement in UMT frameworks, the proposed system follows UMT principles and does not rely on any parallel sentences. In terms of language processing tools and digitized resources, Manipuri language is still in a nascent stage as compared to other major Indian languages. As Manipuri and English are two distant languages with different language families and different writing styles, even the state-of-the-art unsupervised statistical MT (USMT) and unsupervised neural MT (UNMT) models do not provide reasonable MT performance. Poor MT performance between Manipuri and English is contributed by various factors such as lack of (i) suitable language resources, (ii) an effective translation alignments between Manipuri and English, (iii) an effective tools for processing complex morphological structures, etc. In this study, we first investigate preliminary responses of state-of-the-art USMT (Monoses) [12] and UNMT frameworks, namely, XLM [42], MASS [224] and the system proposed in [14], and identify the following factors affecting the performance:

1. Need to handle sub-word level text processing to capture morphological variation.

2. Need of developing an effective phrase-table.

Motivated by the above influencing factors, in this chapter, we propose a Manipuri-English MT system by incorporating appropriate sub-word level pre-processing, transliteration pairs, and document level translation features to a state-of-the-art USMT model, namely Monoses [12] and made the following major contributions.

- Create a news domain Manipuri-English MT test dataset to track the progress of this important field.
- Propose a method to address data sparsity due to agglutinative nature of Manipuri text over a limited dataset by using a suffix segmenter.
- As the digitized bilingual dictionary (required for generating cross-lingual embeddings) between Manipuri and English is not readily available, this study presents two different methods for incorporating transliteration features to exploit phonetically similar transliterated words (loanwords and named-entities).
- Propose an approach to improve the USMT model by utilising document-aligned comparable corpus. Our proposed methodology can take advantage of document level alignments that provide only weak indications of translation equivalence on their own.

6.2 Related Studies

There are only a few thousand publicly accessible parallel sentences of varying domains [94, 18, 75], which is not enough for the standard SMT and NMT. Other

methods developed explicitly for low-resource MT like pivot-based [233], back-translation [58], incorporating a separate language model [199], etc. are also not feasible considering the size of the available dataset. Although a few authors have reported the development of Manipuri-English MT systems [214, 203, 209] using in-house generated training sentences, these datasets are also small and publicly unavailable. UMT could be the alternative solution without the need for the sentence-level parallel corpus. This study is the first attempt for developing MT for Manipuri-English languages pair without using any parallel sentences. To the best of our knowledge, study in [207] is the only available literature for Manipuri UMT. The authors created a Manipuri-English UNMT based on a transformer with a shared encoder and language-specific decoders, similar to the architecture in [14]. They use a few parallel sentences as a development set to fine-tune the models. However, unlike their study, we do not use parallel sentences and address critical language pair specific issues.

As discussed in Section 2.3, studies related to UMT can be broadly grouped into two approaches; (i) USMT and (ii) UNMT. At the core, both approaches depend highly on cross-lingual embeddings (CLEs) between the source and target languages. The USMT models in [127, 14, 13] follow the standard SMT [115] approach with a log-linear combination of translation model, word/phrase penalty, language model, etc. However, the most important component of the translation model, the phrase-table, is obtained in an unsupervised fashion by exploiting the CLEs. UNMT models reported in [14, 126] use the unsupervised CLEs to initialize the model following the NMT paradigm,. The models are then enhanced by using denoising auto-encoder and iterative back-translations. A detailed description of USMT and UNMT models are presented below.

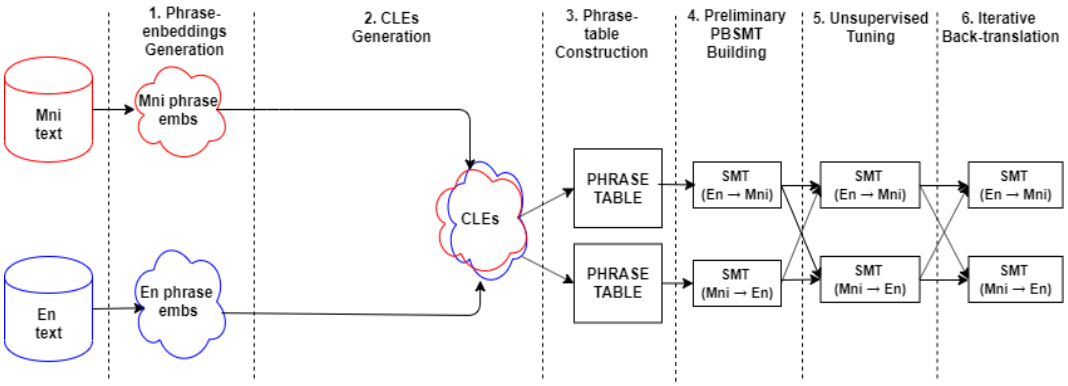


Figure 6.1: A systematic block diagram of the USMT architecture.

6.2.1 Unsupervised Statistical MT

Motivated by the phrase-based SMT models [115], USMT models follow multi-step modular architecture. In this section, we describe the state-of-the-art USMT, the Monoses, in detail. It may be noted that other recent USMT models [127, 13, 143] are also adapted conceptually from Monoses. The Monoses consists of six major steps, as shown in Figure 6.1:

1. *Generating phase-embeddings from the source and target languages independently* using an extension of the standard word-based skip-gram model [149] to n-grams, called phrase2vec* [14]. We considers uni-gram, bi-gram, and tri-gram embeddings following previous studies [14, 13].

2. *Map the source and target language embeddings to a shared cross-lingual embedding (CLE) space* [11]. A detail description of the mapping method is presented in Section 6.5.1.

3. *Construct initial phrase-table* by extracting 100 nearest-neighbors phrases from target language ($\bar{t}$) for each phrase in source language ($\bar{s}$) over the CLEs

*https://github.com/artetxem/phrase2vec

space. The phrase translation probabilities φ_{ph} are computed as follows:

$$\varphi_{ph}(\bar{t}|\bar{s}) = \frac{exp(cos(\bar{s},\bar{t})/\tau)}{\sum_{\bar{t}'} exp(cos(\bar{s},\bar{t}')/\tau)} \tag{6.1}$$

The temperature τ is used to control the confidence of the predictions as in [14]. Along with φ_{ph}, lexical weights are also estimated as follows:

$$lex(\bar{t}|\bar{s}) = \prod_i max(\varepsilon, max_j \varphi_w(t_i|s_j)) \tag{6.2}$$

where the value of ε is fixed at 0.3 to guarantee a minimum similarity score [13]. Word translation probabilities φ_w are also computed in the same manner as that of the φ_{ph} but in the word-level. It is given by:

$$\varphi_w(t|s) = \frac{exp(cos(s,t)/\tau)}{\sum_{t'} exp(cos(s,t')/\tau)} \tag{6.3}$$

The inverse phrase and lexical translation probabilities entry to phrase table are calculated analogously to the phrase and lexical translation probabilities.

4. *Build preliminary phrase-based SMTs (PBSMT)* [115] in both the translation directions using the initial phrase-table, word/phrase penalty, and language model. (5) The preliminary PBSMT models are then tuned iteratively in an unsupervised setting through back-translation on a non-parallel development dataset. Specifically, one of the two initial PBSMT models is utilise to construct a synthetic parallel corpus through back-translation, then apply MERT to tune the model in the opposite direction, iterating until convergence. We consider a random subset of 10,000 non-parallel sentences from

the source and target corpora as a development set for tuning process.

5. Finally, the fine-tuned USMT model undergo rounds of *iterative back-translation*. The purpose of iterative back-translation is to transform the unsupervised setting into a supervised one by exploiting the reverse SMT model. Specifically, initial UMT systems generate synthetic parallel corpus through back-translation and use it to train a conventional phrase-based SMT iteratively.

6.2.2 Unsupervised Neural MT

UNMT models [127, 14, 224, 42] generally follows encoder-decoder setup following the NMT paradigm. Earlier UNMT models initialise the encoder-decoder architecture with unsupervised cross-lingual embeddings (CLEs) [127, 14]. Lample et al. [126] use a single encoder and a single decoder for both the source and target languages. Artetxe et al. [14], on the other hand, utilize a shared encoder but two independent decoders. The models adapt denoising language modeling [86] to enables the model to reconstruct sentences of both the languages. Here, artificial noises (word deletion or permutation) are injected into a clean sentence to create a corrupted input. The denoising objective is set to reorder noisy input into proper syntax, which is necessary for producing fluent outputs. This is done with monolingual data for each language separately. Finally, back-translation [199] is deployed in the training procedure to train UNMT systems for both the translation directions without breaching the constraint of using only monolingual corpora. Specifically, given two input sentences (x, y) from the two languages, two synthetic sentence pairs $x \rightarrow \hat{y}$ and $y \rightarrow \hat{x}$ are obtained via the initial models. The model then learns the translation task on both the language sides by using the reversed

sentence pairs $\hat{y} \rightarrow x$ and $\hat{x} \rightarrow y$ in the NMT scenario. The model may be too weak to generate appropriate translations in the early phases of training. As a result, most techniques update the training data as the model improves over time. The improved source-to-target direction model back-translates source monolingual data, which improves the target-to-source direction model, and vice versa.

Motivated by the recent success of pre-trained language models for various language understanding and generation tasks, several authors have considered pre-training UNMT encoder and decoder on monolingual data [50]. Conneau & Lample [42] in their proposed model, the XLM (Cross-lingual language model pre-training), initialize the encoder and decoder with separate language models trained by a combination of cross-lingual language model pre-training techniques [50]. However, one disadvantage of pre-training the encoder and decoder separately is that it is difficult to train the encoder-decoder attention, which is critical in NMT for inter-connecting the source and target language sentence representations. To counter this limitation, Song et al. [224] proposes MASS which is mAsked sequence to sequence pre-training model. The model randomly masks several consecutive tokens in the encoder's input sentence and predicts the masked fragment in the decoder. This enables the model to jointly pre-train each component in NMT architecture to simultaneously learn to understand the input sentences and improve the translation performance. MASS has outperformed XLM and attention-based NMT model[16] with a BLEU score of 37.5 for English-French MT.

6.3 Empirical investigation of previous Unsupervised MT approaches

The capability of the UMT models to an actual low-resource scenario is still in question. Previous UMT-related studies [127, 14, 42, 224] are predominantly investigated on combinations of high resource languages. For such languages, standard SMT [115] and NMT [16] generally works well, and quality monolingual corpora are also available in abundance [12]. On the other hand, studies in [143, 131] have reported that USMT and UNMT performances usually vary based on the similarity/difference of the source and the target language characteristics like quantity and quality of bilingual corpus, language branch, alphabet, morphology, etc. Not only Manipuri lacks a large-quality monolingual corpus, but the language is also very different from English [40]. The previous study related to unsupervised Manipuri-English MT has only exploited UNMT models [207]. However, when considering resource-scarce languages, statistical machine translation (SMT) generally outperforms neural machine translation (NMT) [55].

Motivated by the above reason, investigating the performances of both the USMT and UNMT models on the distant language pair is meaningful and challenging. To the best of our knowledge, this study is the first attempt to investigate the performance of the USMT model on Manipuri language. This study performs a preliminary investigation of the responses of XLM, MASS, CLE-based UNMT [14] and Monoses [14], a USMT model.

Table 6.1: English-Manipuri News Domain Comparable Corpora.

Language	Documents	Sentences	Words	Vocabulary
English	13408	136560	5.79M	80855
Manipuri	13117	273108	5.62M	277406

Table 6.2: English-Manipuri News Domain MT Test Data.

	Sentences	Tokens for Reference-1	Tokens for Reference-2
English	1006	13040	13412
Manipuri	1006	11168	11123

6.3.1 Experimental Setup

I. Dataset

We consider our Manipuri-English comparable corpus generated from news articles published on *Sangai Express*[*] and *Poknapham*[†], presented in Chapter 3. The Moses Tokenizer[‡] is used to tokenize lower-cased English texts, while a simple white-space tokenization scheme is used for Manipuri texts[§]. Table 6.1 contains a detailed description of the corpus.

To evaluate different MT systems, we manually generate a news-domain MT testset consisting of *1006 parallel sentences*. For a given sentence, we may have multiple possible valid target translation. Therefore, we manually generate two reference translations for each source sentence. The test sentences are subjected to manual quality checks. We followed the setup presented in the study [72]. We asked two different annotators to rate sentence pairs by giving a score between 0 and 10. In our guideline, the 0 score represents a translation that is entirely incorrect and inaccurate, while the 10 represents a perfect translation. We took

[*] https://www.thesangaiexpress.com/
[†] http://www.poknapham.in/
[‡] https://github.com/moses-smt/mosesdecoder
[§] Punctuation symbols are first normalized.

the average score for each sentence pair and rejected translations whose scores were below 7. To ensure consistency, we also rejected pairs in which the difference in the scores among the annotators was above 3 points. A description of the testing dataset is given in Table 6.2. All the models discussed in this chapter are tested on these datasets.

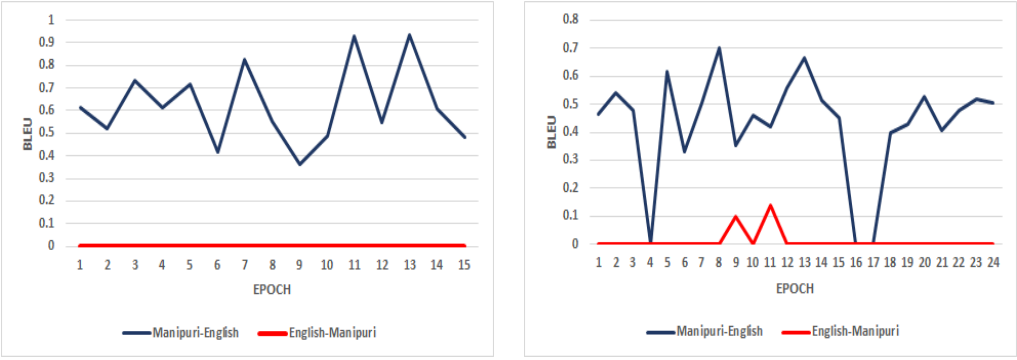
II. Unsupervised MT Configurations

Monoses* follows the same configuration settings as in the original work [14]. Sentences with less than three tokens or more than 80 tokens are deleted, and the rest of the sentences are shuffled for training the CLEs. We use phrase2vec† to generate pre-trained unigram, bigram, and trigram phrase embeddings. The 5-gram language model is estimated using the KenLM [81]. We use MERT [163] for unsupervised tuning. The hyperparameters of XLM‡ and MASS§ are also set based on the studies in [42] and [224] respectively. We use a transformer setting with a 6-layer encoder and decoder to pre-trained and fine-tuned the models for Manipuri-English MT. The embedding size is fixed to 1024. We jointly learn 60k sub-word units between source and target languages using BPE [199]. The model is fined tuned by using Adam optimizer [110] with an initial learning rate and a batch size of 10^{-4} and 500 respectively. However, unlike the studies [224, 42] which use multiple GPUs, we use only a single GPU with 12GB memory for training the models. For the CLE-based UNMT model [14], we consider the original implementation¶ and default settings. We use the skip-gram model with ten negative

*https://github.com/artetxem/monoses
†https://github.com/artetxem/phrase2vec
‡https://github.com/facebookresearch/XLM
§https://github.com/microsoft/MASS
¶https://github.com/artetxem/undreamt

Table 6.3: Translation results of previous UMT models.

Models	*En* → *Mni*		*Mni* → *En*	
	BLEU	**ChrF++**	**BLEU**	**ChrF++**
XLM [42]	≈0	3.26	≈0	2.80
MASS [224]	≈0	2.90	0.48	6.87
Artetxe et al. [14]	4.12	25.67	5.58	23.62
Monoses [12]	4.97	26.78	6.07	23.71



(a) Tuned (b) Pre-trained

Figure 6.2: The performances of MASS during fine-tuning and pre-training.

samples to generate monolingual embeddings with size 300.

All the MT models are evaluated using: (1) BLEU [168]*, and (2) ChrF++ [171]†. ChrF++ computes the F-score averaged on all character and word n-grams. We consider the default word n-gram order of two and character n-gram order of six that have shown to correlate better with direct human assessments. Details regarding the evaluation matrices are presented in Appendix B.

6.3.2 Results and Discussion

Table 4.4 present the results of previous UMT models for both the translation directions: English-to-Manipuri (*En*→*Mni*) and Manipuri-to-English (*Mni*→*En*).

*https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

†https://github.com/m-popovic/chrF

It is evident from the results that MASS and XLM fail miserably for the language pair with a BLEU score of close to 0% in both translation directions. To further confirm pre-training based UNMT models low performance, we evaluate MASS at the end of each epoch during training. Figure 6.2 (a) shows the progress of the model in terms of BLEU score during fine-tuning. It is found that the model never gets going and remains hovering between 0.4 to 0.9 BLEU score for *Mni* $\rightarrow$ *En*, while the *En* $\rightarrow$ *Mni* score remains static at zero. Even during the pre-training stage, the model fails to advance, as shown in Figure 6.2 (b). Similar findings were also previously reported for several distant language pairs [108], including Manipuri-English pair [207]. Apart from the issues with distant language pairs, we believe that the small size of the training corpus may also contribute to the low BLEU and ChrF++ scores. In previous studies [42, 224], these models have typically been trained on huge corpora (in terms of billions of words). However, such resources are currently unavailable for Manipuri. Monoses and CLE-based UNMT [14], on the other hand, performs relatively better than the other UNMT models. Monoses obtains the best BLEU score of 4.97 for *En*$\rightarrow$*Mni* and a score of 6.07 for *Mni*$\rightarrow$*En* outperforming all the UNMT systems. Similar results are also observed for the ChrF++ evaluation matric. This shows that compared to the end-to-end design of UNMT models, the modular architecture of the USMT model is better suited for the language pair.

Although Monoses provides promising performance, there are still lots of translation errors. A manual investigation of the translation results reveals that a significant part of the translation errors is due to morphological variations, and problems in phrase-table mapping resulted from poor source and target translation phrases alignments. Motivated by these observations, this study proposes

to incorporate sub-word level processing to reduce the problem of morphological errors and enhance phrase-table by exploiting a document-aligned comparable corpus and entities/loanwords level transliteration.

6.4 Incorporate sub-words information using Suffix Segmenter

Words in Manipuri are primarily associated with suffixes depending on the number, gender, etc. [161]. Suffixes are more prominent than the prefixes, while there are no infixes [161]. Such inflections produce huge vocabulary leading to many unseen and low-frequency words. Exploiting sub-word information has always assisted the conventional SMT and NMT when dealing with morphologically rich languages. UNMT models have also exploited Byte-Pair Encodings* (BPEs). With similar motivation, we consider a Manipuri suffix segmenter to segment words into root and its suffixes to alleviate data sparseness due to the morphological inflections. Specifically, we use our proposed *Manipuri suffix segmenter*, presented in the Section 5.4, as a pre-processor to segment Manipuri texts before training the model. For example, Manipuri words like মণিপুরগী (for Manipur), মণিপুরদগী (from Manipur), মণিপুরদা (to Manipur), etc. are segmented by separating the root মণিপুর (Manipur) and its suffixes গী , দা and দগী, thereby keeping both the word's principal meaning associated with the root and the auxiliary purposes carried by the suffixes intact.

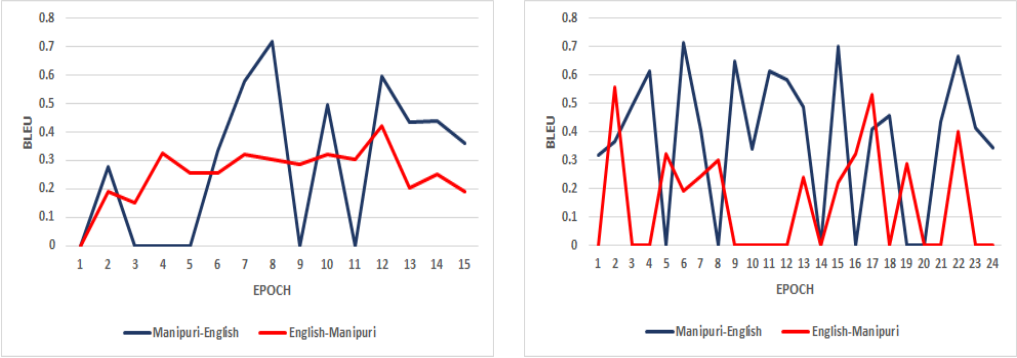
6.4.1 Results and Discussion

To investigate the effectiveness of the proposed suffix segmenter for MT, we compare the performance of the previous models (XLM, MASS, CLE-based UNMT [14]

*https://github.com/glample/fastBPE

Table 6.4: Translation results over non-segmented and segmented corpora.

Models	*En* → *Mni*				*Mni* → *En*			
	BLEU		ChrF++		BLEU		ChrF++	
	Non-seg	**Seg**	**Non-seg**	**Seg**	**Non-seg**	**Seg**	**Non-seg**	**Seg**
XLM	≈0	0.16	3.26	5.45	≈0	0.18	2.80	5.21
MASS	≈0	0.19	2.90	5.33	0.48	0.36	6.87	6.01
Artetxe et al. [14]	4.12	5.85	25.67	27.01	5.58	5.42	23.62	23.62
Monoses	4.97	**6.10**	26.78	**28.01**	6.07	**7.78**	23.71	**27.40**



(a) Tuned (b) Pre-trained

Figure 6.3: The performances of MASS during fine-tuning and pre-training on segmented dataset.

and Monoses [12]) on the *non-segmented* (presented in Table 6.1) and the *segmented* corpora. The *segmented corpus* is obtained after applying the suffix segmenter on the Manipuri text given in non-segmented corpus. Table 6.4 presents the translation results on de-segmented outputs for both the translation directions. The performance of the models on the pre-processed corpus obtained after applying the suffix segmenting algorithm is denoted by *Seg*. It is clearly evident from the table that the BLEU and ChrF++ scores for both the translation directions increases on the segmented dataset in almost all the cases, except for the underperforming model, the MASS and UCLE-based UNMT, in *Mni*→*En*. In the case of Monoses, the translation results on the segmented dataset for *En*→*Mni* and *Mni*→*En* MT increases by about by 1.13 and 1.71 BLEU points, respectively, over

the non-segmented dataset. The results clearly demonstrate that segmenting suffixes improve overall performance by reducing data sparseness. However, similar to the observations obtained on the non-segmented corpus, the state-of-the-art UNMT models, the MASS and the XLM, still fail to perform. Figure 6.3 (a) and (b) shows the progress of the model during tuning and pre-training phase on the segmented corpus. This further validates the inability of the models to capture translation features for Manipuri-English language pair.

6.5 Incorporating Transliteration Features

Previous studies have used shared vocabularies (named-entities, loanwords, etc) to obtain inter-language connection points between the source and the target languages [42, 224, 13]. For Manipuri-English MT, usage of shared vocabulary is not feasible because of different writing scripts. Word replacement/bilingual dictionary approaches [227, 57] are also not suitable because of the lack of the language processing tools/digitized resources like POS/NER taggers, Manipuri-English bilingual dictionary. Under the given circumstances, exploiting phonetically similar transliteration pairs is an encouraging approach without the need of the above resources. Further, consider the nature of our MT dataset, comparability between source language corpus and target language corpus also help in finding matching words.

To generate transliteration features, we rely on transliteration model (TM). TM converts a word in source language to the script of the target language by maintaining the phonetic characteristics of the source language. For example, the word *Imphal* is transliterated as ইম্ফাল in Manipuri script. We consider

bi-directional GRU encoder-decoder grapheme-based transliteration model setup, presented in Chapter 4. The task of an encoder is to understand the character sequence $x_1, x_2, x_3, ..., x_n$ of the input word and decoder is responsible for generating the output word character sequence $y_1, y_2, y_3, ..., y_m$ [67].

The models are trained using the same dataset published in our transliteration study [123], given in Chapter 4. The dataset consists of 4428 training transliteration pairs (TPs), 1000 development pairs, and 607 testing pairs. We maintain a learning rate of 0.001 and batch size of 32. The size of the hidden layer is fixed to 512 and embedding dimension to 256. The model achieve a character accuracy (CA) of 92.66% and 88.35% for English-to-Manipuri and Manipuri-to-English transliteration, respectively. Considering that we do not need to consider every transliteration pairs (but only few set of transliteration pairs) in our proposed model, the obtained accuracy is reasonable for the task. Further, as observed in [235], if the CA of the transliteration pair is above a threshold, the transliteration pair may be considered as matching pair.

After generating the transliteration feature, the next question is how to incorporate these features. For this, we exploited the modular design of USMT. Specifically, we propose two novel extensions to incorporate transliteration features: (1) Improving CLEs by exploiting automatically generated transliteration pairs (TPs), and (2) Improving phrase-table using transliteration models.

6.5.1 Improving CLEs by exploiting automatically generated Transliteration Pairs:

CLEs form the core of the USMT as they are directly used to obtain the phrase-table of the initial model (as discussed in Section 6.2.1). These CLEs are obtained

by mapping the source (X) and the target language monolingual phrase embeddings (Z) to a common embedding space, where the translations are close to each other in the shared space. Specifically, the goal is to learn orthogonal transformation matrices W_X and W_Z such that the cosine similarity of the words/phrases that are translations of one another is maximized over the initial seed dictionary matching matrix D:

$$\hat{W}_X, \hat{W}_Z = \arg\max_{W_X, W_Z} \sum_{X_i \in V_X} \sum_{Z_j \in V_Z} (D_{i,j}((X_i W_X) \cdot (Z_j W_Z))) \tag{6.4}$$

where V_X and V_Z are the phrase vocabulary set of the source and target languages, respectively. Monoses induces the initial bilingual dictionary (D) by considering word frequency distribution of the source and target monolingual corpora using the CLE method proposed in [11] (Step 2 in Figure 6.1). The idea is that the most frequent words in the source corpus may share a semantic relationship with those frequent words in the target corpus. This assumption is not true for the Manipuri and English pair. It is also evident from Monoses' poor translation performance in our preliminary investigation. In our study, instead of a frequency-based induced bilingual dictionary, we use *transliteration pairs* that exceed character accuracy (CA) threshold T_θ as follows. Specifically, a cell $D_{i,j} = 1$ if $\text{CA}(x'_i, z_j) \geq T_\theta$ and $\text{CA}(x_i, z'_j) \geq T_\theta$, where words $x_i \in V_X$ and $z_j \in V_Z$. x'_i and z'_j represents the transliterated versions of x_i and z_j respectively towards the other language using the transliteration models. Otherwise $D_{i,j}$ is set to 0.

Since, both the transformations are orthogonal, $W_X = U$ and $W_Z = V$ are the optimal solutions of Equation 7.1, where $USV^T = X^T DZ$ is the singular value decomposition of $X^T DZ$ [11]. Similar to the settings in [10], we adopt multi-step

pre-processing: length normalization, mean centering, and whitening. These are followed by the post-processing steps: re-weighting, de-whitening, and dimensional reduction. We further iteratively populate the dictionary matrix using the estimated matrices ($\hat{W}_X$, $\hat{W}_Z$) to create a new seed dictionary (D). The matrix D is populated by using Cross-domain Similarity Local Scaling (CSLS) as proposed in [43]. For each word/phrase pairs x_i and z_j, we update $D_{i,j} = 1$ if the CSLS score of x_i with z_j is highest as compared to other words in V_Z, and $D_{i,j} = 0$ otherwise. The dictionary is induced for both the directions.

6.5.2 Improving Phrase-table using Transliteration Models

Instead of semi-supervising CLEs generation, in this method, we directly incorporate the transliteration scores as generated by transliteration models (TMs) in the phrase-table to enhance phrase alignments (Step 3 in Figure 6.1). Specifically, we re-score the phrase-translation and lexical probabilities using TMs. TMs enable the USMT to consider phonetic similarities between the source phrase embedding ($\bar{s}$) and the mapped target phrase embedding ($\bar{t}$). We investigate three different ways for improving the phrase-table.

1. *Re-score Lexical Weights(RS-lex)*: In this method, we introduce transliteration weights in place of lexical weights. The transliteration weights enable the model to exploit phonetic similarities, and are estimated using the TMs, as follows:

$$tns(\bar{t}|\bar{s}) = \prod_i max(\varepsilon, max_j CA(t_i, TM_{S\rightarrow T}(s_j)) \tag{6.5}$$

Here, $TM_{S\rightarrow T}(x)$ is the transliterated word of the source word x using the source-to-

Table 6.5: Results of using transliteration features along with baselines.

Models	*En* → *Mni*		*Mni* → *En*	
	BLEU	**ChrF++**	**BLEU**	**ChrF++**
Unsupervised CLEs	6.10	28.01	7.78	27.40
CLEs initialize with 25 word pairs	6.18	28.55	7.94	28.57
RS-lex	6.34	28.40	7.97	28.61
RS-phrase	6.31	28.06	7.75	27.32
RS-phrase-lex	6.37	28.42	**8.27**	29.19
CLEs initialize with TPs (CA = 80%)	6.35	28.39	7.89	28.06
CLEs initialize with TPs (CA = 100%)	**6.45**	**28.86**	8.26	**29.27**

target transliteration model (TM), and $CA(x, y)$ represents the character accuracy ([0,1]) between the word x and y. ε is a constant fixed at 0.3 [14].

2. *Re-score phrase translation probabilities (RS-phrase)*: In this case, we modify the phrase translation probabilities φ_{pb} itself by incorporating the transliteration weights $tns(\bar{t}|\bar{s})$ as follows:

$$\varphi_{pb}(\bar{t}|\bar{s}) = \frac{exp(cos(\bar{s}, \bar{t})/\tau)}{\sum_{\bar{t}'} exp(cos(\bar{s}, \bar{t}')/\tau)} * tns(\bar{t}|\bar{s}) \tag{6.6}$$

3. *Re-score both the phrase translation probabilities and lexical weights (RS-phrase-lex)*: In this method, we use the equation 6.6 for estimating the φ_{pb} and equation 6.5 for estimating the lexical weights alternative, the transliteration weights.

6.5.3 Results and Discussion

Table 6.5 show the results of using transliteration features in Monoses over the segmented dataset. We compare the performance of our proposed transliteration features incorporation methods to two baselines to evaluate them properly: 1)

Monoses initialized with unsupervised CLEs* and 2) Monoses initialized with a dictionary consisting of 25 word pairs (most frequent pairs present in the corpus) [11]. The next three rows show results of our proposed phrase-table re-scoring methods, while the final two rows represent the results of using automatically generated TPs as bilingual supervision. We tested with two variants of the initialization scheme for the proposed semi-supervised model:

1. Initialize with TPs generated with the character accuracy (CA) threshold set to 80%.

2. Initialize with TPs generated with the character accuracy (CA) threshold set to 100%.

It is evident from the results that for all the cases, except the *Monoses with RS-phrase* for Mni→En direction, the proposed methods outperform the baselines. Weakly supervising the CLEs using the TPs obtained the best result with 6.45 BLEU for En→Mni. On the other hand, *Monoses with RS-phrase-lex* achieved the best BLEU score for the Mni→En narrowly beating the semi-supervised counterpart by only 0.001 BLEU points. We also observe that the variant with 100% CA consistently outperforms all the baselines proving that providing supervision using automatically generated TPs helps. However, we noticed that TPs generated with 80% CA contain lots of noises. As a result, this variant of initialization hurts the performance. The results clearly show that the proposed methods can exploit the phonetically similar transliterated words between the language pair.

Table 6.6: Examples of target words ranked based on the nearest neighbour retrieval from the CLEs space. The italic words in the bracket represent the corresponding Manipuri word transliteration in the Roman alphabet.

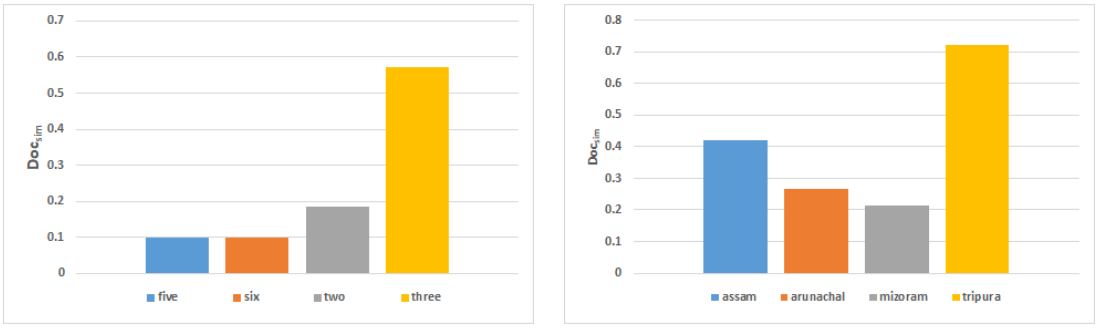
Word	**References**	**Nearest-1**	**Nearest-2**	**Nearest-3**
অহুম (*ahum*)	three	five	six	two
ত্রিপুরা (*tripura*)	tripura	assam	arunachal	mizoram
নোংপোক (*nongpok*)	east	northeast	northeastern	northeast
মুগা (*muga*)	silk	herbal	handicraft	geotextile

6.6 Exploiting Document level alignments

A major limitation of the methods discussed above is that the phrase-table is directly induced from CLEs. In the same way that semantically similar words tend to be closer in monolingual embeddings [149], the condition holds for CLEs. This is because they are generated by mapping the independently trained source and target language embeddings into a shared embeddings space by using linear transformations, where the translations are optimized to be close to each other (discussed in Section 6.5.1). As a result, semantically similar words are often obtained instead of the correct target word/phrase while retrieving nearest neighbours. Table 6.6 show examples of target words ranked based on the nearest neighbour retrieval from the CLEs space for several source words. The table shows that the model failed to identify the correct target word, instead semantically comparable target words are predicted.

As an alternative, this study presents a method that re-score the retrieved phrases in the USMT architecture based on the features derived from document alignments. As document-aligned pairs represent the same topic/event, we hypothesis that source and target word pairs in aligned documents are more likely

*Monoses default setting.



(a) অহুম (b) ত্রিপুরা

Figure 6.4: Examples of semantically similar English words scored using the doc_{sim} for two Manipuri words. Yellow bar represents the correct translation.

to be translations than non-aligned ones. Utilising these characteristics, the model should be able to select the proper target word from a pool of semantically comparable translations. Motivated by the above reasons, we propose an approach that can exploit a small document-aligned comparable corpus. The proposed method is different from previous approaches developed for utilizing comparable corpus. Most of the previous studies attempted to extract parallel segments (sentences, phrases, or words) from comparable corpus to assist conventional data-driven MT approaches [79, 254]. However, apart from a few, most parallel segment extraction algorithms are supervised [228]. Furthermore, a significant number of high-quality comparable corpus is required [245], which is not available for most of the low-resource language pairs.

The proposed approach consist of two phrases. Firstly, we determine a similarity score between source and target phrases based on document alignments. To calculate the similarity score, we create a document vector for each phrase by sorting the documents in the aligned-corpus and counting the number of phrase occurrences in each document. Consider A and B are source (a) and target (b) language phrases document vectors, respectively. Then, the k^{th} entry of A repre-

sents frequency (f_k) of the phrase a in the k^{th} source document. Similarly, the k^{th} entry of a target vector B represents frequency of the phrase (b) in the target document that is aligned with the k^{th} source document. We normalize the vectors by dividing all of f_k components by the total count of the phrase. Then, the similarity between the pair (a and b) is computed as cosine similarity over the normalized document vectors as:

$$doc_{sim}(a,b) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{6.7}$$

Figure 6.4 (a) and (b) shows examples of semantically similar English words scored using the doc_{sim} for the Manipuri words অহুম (three) and ত্রিপুরা (tripura). It is evident for the figure that the correct translation are ranked highly (shown in yellow color) as compare to semantically similar words. This shows the effectiveness of the similarity score in disambiguating semantically similar words.

After calculating the similarity score, the next question is how to incorporate the score in the end-to-end UMT training. To accomplish this, we project the phrase-table induction as a parallel segment scoring problem by borrowing idea from parallel segments extraction methods [151, 2, 247, 222]. However, instead of extracting segments which translation similarity is above a threshold, we assign a score by combining all the translation features. In our case, the 100 most nearest neighbours scores calculated directly from the CLEs (refer Section 6.2.1 Step 3) are combine with (doc_{sim}). We consider the weighted sum method [248] for combining the scores:

$$\varphi'_{ph}(\bar{t}|\bar{s}) = w * \frac{e^{(cos(\bar{s},\bar{t})/\tau)}}{\sum_{\bar{t}'} e^{(cos(\bar{s},\bar{t}')/\tau)}} + (1-w) * \frac{e^{(doc_{sim}(\bar{s},\bar{t}))}}{\sum_{\bar{t}'} e^{(doc_{sim}(\bar{s},\bar{t}'))}} \tag{6.8}$$

Table 6.7: Manipuri-English Document-aligned Comparable Corpus.

Language	Doc-aligned	Sentences	Tokens
English	2658	37626	1.09M
Manipuri	2658	45050	1.34M

where, w is the weight assigned to the original phrase-translation score (φ_{ph}). Following the φ_{ph} computation, we also apply the softmax function over the doc_{sim}. The intuition of using weighted sum is that there is no such thing as a perfectly definite and comprehensive translation features particularly in case of unsupervised/semi-supervised settings. As a result, scores derived from each of the features must be weighted such that the final ensemble reflects the optimum combination. Using the same idea, the lexical translation probability is also obtained as follows:

$$lex'(\bar{t}|\bar{s}) = w * \prod_i max(\varepsilon, max_j \varphi_w(t_i|s_j)) + (1-w) * \prod_i max(\varepsilon, max_j doc^w_{sim}(t_i|s_j)) \quad (6.9)$$

The word translation probabilities doc^w_{sim} are calculated using the same method as doc_{sim}, with the exception that the scoring function is only applied to unigrams.

6.6.1 Experimental Setup

Out of the Manipuri and English documents presented in Table 6.1, we manually aligned few of them (refer Section 3.4) to obtain document level alignments. The document-aligned corpus statistics is shown in Table 6.7. We then segment Manipuri words into root and suffixes using the proposed segmenter.

The proposed method is implemented over the original distribution of Monoses*.

*https://github.com/artetxem/monoses

Table 6.8: Results of the proposed method exploiting document alignment characteristics along with baselines.

Models	*En* → *Mni*		*Mni* → *En*	
	BLEU	**ChrF++**	**BLEU**	**ChrF++**
Monoses [14]	6.10	28.01	7.78	27.40
RS-phrase-lex	6.37	28.42	**8.27**	29.19
CLEs initialize with TPs (CA = 100%)	6.45	28.86	8.26	29.27
Proposed Model ($w = 0.5$)	6.21	28.56	7.99	29.53
Proposed Model ($w = 0.7$)	**6.71**	**29.35**	**8.80**	**29.86**
Proposed Model ($w = 0.9$)	6.53	29.02	8.35	29.61

The model is initialize with transliteration pairs generated with the character accuracy (CA) threshold set to 100%. All the configurations of our proposed model are set the same as the other models discussed above to make the systems comparable. These models are also evaluated on the same testing data given in Table 6.2.

6.6.2 Results and Discussion

Table 6.8 show the experimental results for both the translation directions. The first three rows represent the performance of the baselines: (1) Default Monoses, (2) Monoses with RS-phrase (Re-score both the phrase translation probabilities and lexical weights), and (3) Monoses initialize with transliteration pairs generated with the character accuracy(CA) threshold set to 100%. We consider three different settings of our proposed model ($w = 0.5$, 0.7 and 0.9) to better understand the effect of each features. It is evident from the results that the proposed model is able to capture the document alignment features. We observe that the proposed model with $w = 0.7$ and $w = 0.9$ outperform the baselines for both the translation directions. The proposed model ($w = 0.7$) obtain the best performance with 6.71 and 8.80 BLEU points for the *En* → *Mni* and *Mni* → *En*, respectively. Similar improvements are also observed for the ChrF++ evaluation metric. The

results also confirm that the proposed model can exploit the different translation features to disambiguate the semantically similar words. If we compare the different settings of the proposed model, we found that with change in the parameter w, the translation results also vary. The setting with $w = 0.5$ under-perform the baseline models. This shows that out of the two used features, the CLEs based feature is more effective than the document-aligned features.

6.7 Summary

This chapter developed an MT system for Manipuri-English language pairs without parallel sentences. We showed that a relatively cheaper comparable corpus could be considered a potential alternative over expensive parallel sentences for MT task, even for distant languages. This study also provided some necessary modifications on the popular USMT model, the Monoses, to adapt for the Manipuri-English language pair. Specifically, we incorporated a Manipuri suffix segmenter to reduce the data sparseness due to the agglutinative nature Manipuri language. Furthermore, we proposed two novel methods that enhanced the model by using: (i) transliteration features, and (ii) document-aligned comparable corpus. We bring new insight into the USMT architecture by taking advantage of its modular design to exploit different translation features other than those obtained from CLEs. Moreover, our experimental settings are more realistic and practical than those employed in prior research. We show that a small number of document pairs can also be used for MT in a low-resource situation instead of relying on many document-aligned pairs.

7

Improving Manipuri-English MT by Exploiting a Temporally aligned Comparable Corpus

In Chapter 6, we have developed an MT model for the Manipuri-English language pair without using any parallel sentences. The proposed model can normalize morphological infection issues of Manipuri and incorporate transliter-

ation features in the unsupervised statistical MT architecture. Furthermore, we have enhanced the proposed model by exploiting a small document-aligned comparable pairs. Though the likelihood of finding translated word-pairs from the document-aligned corpus appears to be higher, it does not significantly improve performance due to data sparsity. To address this issue, this chapter outlines a strategy for exploiting a temporally aligned comparable corpus having a broader coverage and availability in place of the expensive document-aligned comparable corpus. This research may be viewed as an extension of our Manipur-English MT model, with suffix segmenter and transliteration elements added (published as a workshop paper in LowResMT 2021 [124]

7.1 Introduction

Scarcity of parallel sentences has always been a major issue for low-resource MT. Although comparable corpora* have been widely employed as a supplementary resource to MT in low-resource situations [179], most of these studies only attempted to extract parallel segments (sentences, phrases, or words) from comparable corpus to aid traditional data-driven MT [79, 254]. As a result, they require a large corpus to extract sufficient number of parallel segments/sentences to train a supervised MT from scratch [245]. On the other hand, recently proposed unsupervised MT (UMT) models: Unsupervised Statistical Machine Translation (USMT) [127, 14] and Unsupervised Neural Machine Translation (UNMT) [224, 42] present a mechanism to develop an end-to-end MT systems without using any parallel sentences. However, such methods still rely heavily on cross-lingual embeddings (CLEs) de-

*Comparable corpora are bilingual texts that are not precise translations of one another, but are aligned at different levels based on some shared characteristics like topic, domain, time, thematic, etc.

rived from monolingual data, which are ineffective when the source and target languages are not comparable (at least in terms of the domain) [143]. Although recent research has examined employing comparable corpus for building UMT systems [108], they fail to explicitly leverage the source and target language corpora's comparable characteristics.

Although MT research has been going on for years, the condition for the Manipur-English language pair is still in its infancy due to the lack of a sufficient number of parallel sentences and language processing tools. Moreover, the lack of a sizeable comparable corpus presents a unique issue for the language pair [207, 124]. On the other hand, our study (discussed in the previous chapter) developed an MT system for the language pair without relying on any parallel sentences by enhancing USMT model. Although, the results are promising, one major disadvantage of the approach is that the model suffer from data sparsity due to the limited document-aligned corpus. Wikipedia articles which acts as a primary source for document-aligned comparable corpus generation, are limited for Manipuri*.

In this study, we exploit the temporally aligned characteristics of our proposed comparable corpus. Temporally aligned corpora are relatively easier to obtain than document-aligned data, as news publications are always associated with date of publications. The method is also motivated by the fact that words that are translations of one another appear in both the source and target languages with similar frequency distributions over time. This is because local news articles in various languages will cover the same or similar events on the same or nearby days. We exploit this characteristic to further enhance the Manipuri-English MT

*The condition is true for most of the low-resource languages.

performance. Contributions make in this chapter are summarized below.

- Propose a temporal cross-lingual embedding method by exploiting transliterated word pairs and temporally aligned comparable corpus.
- Enhance Manipuri to English MT and English to Manipuri MT by 59.80% and 70.62% respectively in terms of BLEU score as compare to Monoses without using parallel sentences.

7.2 Related Studies

Over the years, several approaches have been proposed to utilize comparable corpus for MT. The vast majority of them follows a more traditional approach to the low-resource MT problem. They aim is to first extract sentence/phrase translations from the corpora. The retrieved parallel segments are then used as training examples for data-driven MT systems (SMT and NMT) [2, 247, 222]. However, parallel data extraction from comparable is not a trivial task, especially for low language settings. The downside of such strategies is that they generally require a large parallel sentences to be trained [2, 247, 222, 151, 194, 25, 31, 195, 15]. As a result, these approaches are not feasible for our case. Few recent papers [78, 79, 122, 106] have created unsupervised methods for mining parallel data. Unfortunately, these methods still rely on a robust cross-lingual embeddings [11, 43].

UMT models, on the other hand, belong to a special class of MT systems that depend only on source and target monolingual corpora. However, they can be directly adapted to utilize comparable corpus, which has proven to be more effective [107]. Unlike parallel segments extraction methods, these approach provide a mechanism to train an end-to-end MT model without utilising any parallel

sentences. UNMT generally considers encoder-decoder architecture following the NMT paradigm and utilizes a three-step training process: Initialization, Denoising Auto-Encoder, and Back-translation [127, 14]. USMT, on the other hand, follows the traditions log-linear combination of several models, including translation model, language model, word/phrase penalty, etc. However, the major difference with the convention SMT is that in USMT, these models are learned without using parallel sentences [14, 127, 13]. In the previous chapter, we have already shown that because of the unavailability of a sizeable comparable corpus and distinctive linguistic features between English and Manipur, the modular design of USMT models is better suited for the language pair and outperforms UNMT models. We further enhanced the state-of-the-art USMT model, the Monoses [14], by segmenting Manipuri text into root and suffixes to normalized agglutinative nature of Manipuri. The study also presents a methodology to incorporate transliteration features that further enhance the translation performance [124]. Furthermore, translation features derived from document-level comparable corpus alignment characteristics are also incorporated into the USMT model. However, translation features derived from the document-aligned corpus are not effective enough because of data sparsity. As a result, instead of the document-aligned corpus, we present a method to exploit sub-corpora level alignments, specifically the temporal alignments in this chapter. A description of the proposed model is presented in the subsequent section.

7.3 Proposed Model

Our proposed model can be seen as an enhancement of the Manipuri-English MT model [124] (also presented in the previous Section 6.5.1). The model follows the prominent USMT model, the Monoses [14], setup. The training pipeline of the model [124] is listed below.

1. Phase-embeddings are generated for the source X and target Z languages independently using phrase2vec* [14].

2. The generated source and target language embeddings are mapped to a cross-lingual embedding (CLE) space [11]. Precisely, orthogonal transformation matrices W_X and W_Z is learn to map X and Z into a shared embedding space over the seed dictionary matching matrix (D). The initial dictionary D is obtain by using automatically generated *transliteration pairs*. (refer Section 6.5.1 for details). The objective is to maximize the following function [10]:

$$\hat{W}_X, \hat{W}_Z = \arg\max_{W_X, W_Z} \sum_{X_i \in V_X} \sum_{Z_j \in V_Z} (D_{i,j}((X_i W_X) \cdot (Z_j W_Z))) \tag{7.1}$$

   This training process is iteratively refined by using the estimated matrices ($\hat{W}_X$, $\hat{W}_Z$) to create a new seed dictionary (D). The new D is generated based on the Cross-domain Similarity Local Scaling (CSLS) [43]. Specifically, for each word pairs (X_i, Z_j), we update $D_{i,j} = 1$ if the CSLS score between them is the highest over all combinations of X_i and other target words.

*https://github.com/artetxem/phrase2vec

Otherwise, $D_{i,j} = 0$. The dictionary is induced for both the directions, and then concatenated together [11].

3. Initial phrase-tables are generated by inducing 100 nearest-neighbors target phrases for each phrase in source language phrase over the CLEs space.

4. Preliminary phrase-based SMTs (PBSMTs) [115] are generated for both the translation directions using the initial phrase-table, word/phrase penalty, and language model.

5. The preliminary PBSMT models are then tuned iteratively in an unsupervised setting through back-translation on a random sample of 10,000 non-parallel sentences development dataset.

6. The fine-tuned USMT models are finally subjected to rounds of iterative back-translation.

7.3.1 Incorporating Temporal Alignments

We propose a novel multi-step method to incorporate temporal alignment characteristics to enhance the CLEs (Step 2 of the model discussed above). Figure 7.1 depicts the basic training process of the proposed method. The core idea is to enhance the global cross-lingual embeddings ($X_m = XW_X$ and $Z_m = ZW_Z$) by using the sets of time-specific cross-lingual embeddings that are separately learned under the different temporal-aligned conditions. Inspired by previous works on temporal embeddings under monolingual environments [118, 249] and the successes of CLE models in [11, 43], our temporal cross-lingual embeddings also consider mapping-based approach. Suppose, there are multiple pre-trained set of source language

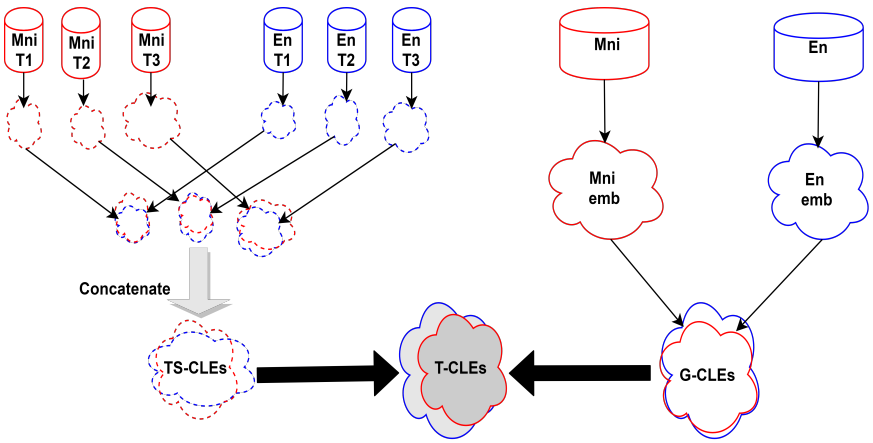


Figure 7.1: A basic diagram representing the generation of temporal-CLEs (T-CLEs) using temporal alignments. G-CLEs represents the global CLEs, while time-specific CLEs are denoted by TS-CLEs.

embeddings X^t and their corresponding target language embeddings Z^t for different time-frames $t = t_1, t_2, ..., t_n$. For each time-frame t_k, we first mapped the source language embeddings and its corresponding target language embeddings to a common space by maximizing and iteratively refining the following objective function (using the same method discussed above):

$$\hat{W}_X^{t_k}, \hat{W}_Z^{t_k} = \arg\max_{W_X^{t_k}, W_Z^{t_k}} \sum_{X_i^{t_k} \in V_X^{t_k}} \sum_{Z_j^{t_k} \in V_Z^{t_k}} (D_{i,j}^{t_k}((X_i^{t_k} W_X^{t_k}) \cdot (Z_j^{t_k} W_Z^{t_k}))) \tag{7.2}$$

where $D_{i,j}^{t_k}$ is the initial transliteration dictionary for the time-frame t^k. The resulting CLEs $X_m^{t_k} = X^{t_k} \hat{W}_X^{t_k}$ and $Z_m^{t_k} = Z^{t_k} \hat{W}_Z^{t_k}$ for each time-frame are expected to preserve local structure across the source and target languages of the particular time t_k. Secondly, we combine all the embeddings $X_m^{t_k}$ by concatenating each entries for all $k = 1, 2, .., n$ into a single file X_m^t. Similarly, the Z_m^t is also obtained.* These time-specific embeddings are then used to enhance the global

*During concatenation, we initially tried by averaging the embeddings of word/phrase that occur in multiple time-frame to keep a single embedding representation for each word/phrase.

Table 7.1: English-Manipuri News Domain Comparable Corpora.

Language	Documents	Sentences	Tokens	Segmented Vocabulary
English	13408	136560	5.79M	80855
Manipuri	13117	273108	5.62M	165998

Table 7.2: English-Manipuri News Domain MT Test Data.

	Sentences	Tokens for Reference-1	Tokens for Reference-2
English	1006	13040	13412
Manipuri	1006	11168	11123

mapped embeddings X_m and Z_m. We achieve this (for the source language side) by learning a linear transformation W^t_X between the global source CLEs X_m and the time-specific target CLEs Z^t_m. The optimal transformation matrix is estimated by solving the Orthogonal Procrustes problem [191] and refining the dictionary D and W^t_X iteratively:

$$\hat{W}^t_X = \arg\min_{W^t_X} \sum_{i,j \in D} ||W^t_X X_m - Z^t_m||_F \tag{7.3}$$

The target language side $\hat{W}^t_Z$ is also obtained analogously. The mapped embeddings $X_m = \hat{W}^t_X X_m$ and $Z_m = \hat{W}^t_Z Z_m$ are finally re-aligned to map them to a common space for generating the desired temporal CLEs by following the mapping procedure discussed above.

7.4 Experimental Setups

To investigate the proposed method effectiveness in exploiting the temporal alignments, we consider the following variant of the Monoses [12] as baselines:

However, this worked poorly in our preliminary experiments. It may be because averaging all the embeddings obtained from different time-frames fail to capture all the time-specific features. The embedding is a particular time may have more influence over another in specifying the overall semantic characteristic of the word/phrase.

Table 7.3: Dataset Description of Temporal Alignments. Months are represented in MM-YY format.

Alignment with 4 (Four) Time-frames		
Months	*English*	*Manipuri*
03-17 to 06-18	1.9M	1.9M
07-18 to 10-18	1.4M	1.5M
11-18 to 08-19	1M	1M
09-19 to 05-20	1.2M	972k
Alignment with 9 (Nine) Time-frames		
Months	*English*	*Manipuri*
03-17 to 12-17	263k	581k
01-18 to 03-18	897k	664k
04-18 to 06-18	770k	689k
07-18 to 08-18	710k	767k
09-18 to 10-18	671k	732k
11-18 to 04-19	495k	549k
05-19 to 08-19	530k	514k
09-19 to 11-19	682k	488k
12-19 to 05-20	587k	483k

1. Monoses initialized with phrase source and target language embeddings before projecting to the cross-lingual space.

2. Monoses initialized with unsupervised CLEs*

3. Monoses initialized with a dictionary consisting of 25 word pairs (most frequent pairs present in the corpus) [11].

4. Monoses initialize with transliteration pairs (TPS) with character accuracy (CA) threshold set to 100%.

These models are trained on our comparable corpus given in Table7.1. Similar to the training pipeline in [11], source and target language corpora are pre-

*Monoses default setting.

processed separately. Moses Tokenizer* is used to tokenize lower-cased English texts, while a simple white-space tokenization scheme is used for Manipuri texts†. Sentences with less than three tokens or more than 80 tokens are deleted, and the rest of the sentences are shuffled for training the CLEs. Following our previous study [124], Manipuri text are segmented into root and suffixes to normalized the morphological inflection issue of the language. *Segmented vocabulary*, shown in Table 7.1, is the number of unique tokens obtained after applying the suffix segmenter.

The temporal alignments are obtain by aligning the corpus, presented in Table 7.1, at the sub-corpora level using the month of publication as a criterion (refer Section 3.4 for the alignment procedure). We consider two different temporal alignments, as shown in Table 7.3: (1) *Temporal alignment with four time-frames*, and (2) *Temporal alignment with nine time-frames*. We also segment the Manipuri text in the temporally aligned corpus. The duration of time-frames is set to ensure that each one contains a roughly equal quantity of tokens.

Our proposed methods are incorporated over the distributed of the Monoses‡. Other configuration settings are kept the same as the original Monoses [11]. All the baselines also follows the same configurations to make the systems comparable. The 5-gram language model is estimated using the KenLM [81]. We use MERT [163] for unsupervised tuning. Following the common practice, all the MT models are evaluated over the de-segmented translation outputs using: (1)

*https://github.com/moses-smt/mosesdecoder
†Punctuation symbols are first normalized.
‡https://github.com/artetxem/monoses

Table 7.4: Results of using temporal alignments along with Monoses semi-supervised with TPs (CA = 100%).

Models	*En* → *Mni*		*Mni* → *En*	
	BLEU	**ChrF++**	**BLEU**	**ChrF++**
Without CLEs	≈0	4.16	≈0	4.09
Unsupervised CLEs	6.10	28.01	7.78	27.40
CLEs initialize with 25 word pairs	6.18	28.55	7.94	28.57
CLEs initialize with TPs (CA = 100%)	6.45	28.86	8.26	29.27
Temporal Alignments with Four Time-frames	6.62	29.14	8.45	29.32
Temporal Alignments with Nine Time-frames	**8.48**	**30.66**	**9.70**	**30.29**

BLEU [168]*, and (2) ChrF++ [171]†. We consider the test dataset given in Table 7.2 (presented in previous chapter).

7.4.1 Transliteration Model Configurations

For populating the initial mapping dictionary, we consider the same setting presented in Section 6.5.1. A grapheme-based bi-directional GRU encoder-decoder transliteration model [67] is used. The hidden layer's size is set to 512, while the embedding dimension is set to 256. With a learning rate of 0.001 and a batch size of 32, we utilise Adam optimizer [110]. The models are trained using the dataset presented in our study [123]. It consist of 4428 training transliteration lexicon pairs with 1000 development pairs. The models gives a word accuracy of 73.27% for English-to-Manipuri transliteration and a word accuracy of 61.7% for Manipuri-to-English transliteration on the testing dataset.

*https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

†https://github.com/m-popovic/chrF

7.5 Results and Discussion

Table 7.4 presents the translation results in term of BLEU and ChrF++ scores. The first four rows represent the baseline models discussed above. The following two rows show results of our proposed method variants: model enrich by using the temporal alignments with four time-frames and nine time-frames, respectively (refer Table 7.3 for dataset description). The results show that random source and target embeddings initialization (shown in the first row of Table 7.4) performed very poorly (BLEU score close to zero) for the language pair. This indicates that CLEs are indeed necessary for the model to perform well. It is also evident from the table that the proposed models outperform their baseline systems in both the translation directions for all the cases. The translation results clearly show that our proposed multi-step method effectively exploits the translation features from the temporal alignments and improves the CLEs. Upon comparing the proposed MT performance on two temporal aligned datasets, we found the variant with nine time-frames performs relatively better than the one with only four time-frames. This is because partitioning the corpus into only four time-frames is insufficient to capture the diversity of time-specific embeddings compared to the nine time-frames. We also experiment by dividing the dataset into more time-frames. However, it hurts the performance as the time-specific CLEs fail to capture the correct semantic representations due to the limited corpus size in each frame.

7.6 Qualitative Analysis

Table 7.5 shows the results of experiments for analyzing the effect of methods proposed throughout this thesis. The first row corresponds to the default Monoses.

Table 7.5: Ablation results of the proposed methods.

	En → *Mni*		*Mni* → *En*	
	BLEU	**ChrF++**	**BLEU**	**ChrF++**
Monoses	4.97	26.78	6.07	23.71
+ Segmentation	6.10	28.01	7.78	27.40
+ Transliteration Pairs (CA = 100%)	6.45	28.86	8.26	29.27
with Document Alignments	6.71	29.35	8.80	29.86
with Temporal Alignments (Time-frames = 9)	**8.48**	**30.66**	**9.70**	**30.29**

The remaining rows represent the performance of our proposed modifications of the Monoses. It is evident from the results that each of the proposed modifications enhances the performance for both translation directions. Our proposed model that exploits temporal alignments (represented in the last row of Table 7.5) brings an improvement of about 3.51 and 3.63 BLEU points in *En*→*Mni* and *Mni*→*En* MT directions, respectively. Similarly, we also observe a significant improvement of about 3.88 and 6.58 points in the ChrF++ metric.

If we compare our two proposed methods that exploit two different comparable corpus characteristics, namely, temporal-alignment and document-alignment, the temporal-alignment-based method is more effective in utilizing the proposed corpus. This may be because of the relatively low coverage of the document-aligned corpus compared to the temporal-aligned comparable corpus. Out of the total 13408 English and 13117 Manipuri documents, only 2658 are document-aligned, while the entire corpus is temporally aligned. On a positive note, both the proposed approaches perform the task they are intended to do. Specifically, exploiting the translation features derived from temporal-aligned and document-aligned characteristics of the comparable corpus. However, it is still early to conclude about their competitiveness as the results are biased towards the temporal-alignment methods due to the nature of our dataset. A thorough investigation on other

Table 7.6: Proposed model translation examples showing correct predictions of unigrams and multi-grams. The italic word/phrase below each Manipuri word/phrase represents their transliteration in the Roman alphabet.

	English → Manipuri
Input	in the last years matric examination, out of 25 rank positions holder, two students are from a government school.
Reference-1	হৌখিবা চহীগী মেট্রিক পরিক্ষাগী ফলদা গবর্নমেন্টকী স্কুলগী মহৈরোয় অনীনা পোজিসন হোল্ডর মখোয় ২৫গী মনুংদা চনবা ঙমখি । *houkhiba chahigi metric parikhyagi falda governmentki schoolgi maheiroi anina position holder makhoi 25gi manungda chanba ngamkhi*
Reference-2	হৌখিবা চহিগী মেট্রিক ইগজাম ফলদা গবর্নমেন্ট স্কুল মহৈরোয় অনীনা রেঙ্ক পোজিসন হোল্ডর মখোয় ২৫গী মনুংদা চনবা ঙমখি । *houkhiba chahigi metric exam falda government school maheiroi anina rank position holder makhoi 25gi manungda chanba ngamkhi*
Predicted	হৌখিবা চহিগী matric ইগজাম, মসিগী রেঙ্ক ২৫ positions holder অনী, মহৈরোয় ময়াম অমা দগী গবর্ণমেন্ট স্কুল ।
	Manipuri → English
Input	জেনিমসকি ডাইরেক্টরনা চিয়ার লৌদুনা মেডিকেল সুপরিন্টেন্দেন্টকি বোর্ড রুমদা মীফম অমা পাঙথোকখি । *jnimski directorna chair louduna medical superintendentki board roomda mipham ama pangthokhi*
Reference-1	jnims director chaired a meeting at medical superintendent board room.
Reference-2	jnims director called a meeting at medical superintendent board room.
Predicted	jnims director chaired a medical সুপরিন্টেন্দেন্ট board room in a meeting was also held.

Table 7.7: Proposed model translation examples showing word-order error. Matching colored texts represent translation equivalents. The italic word/phrase below each Manipuri word/phrase represents their transliteration in the Roman alphabet.

	English → Manipuri
Input	it is good that the chief minister n biren took note of concerns raised by desam
Reference-1	ডেসামনা পুখৎলকখিবা ৱাফম চীফ মিনিষ্টর এন বিরেননা হকচিন্না লৌখিবা অসিমক অফবা ওই *desamna pukhatlakkhiba wapham chief minister n birenna hakchinna loukhiba ashimak aphaba oi*
Reference-2	দেসামনা পুখৎলকপা ৱাফম অদু চীফ মিনিস্তর এন বীরেননসু হকচিন্না লৌখিবা মসি য়াম্না ফৈ *desamna pukhatlakpa wapham adu chief minister n birennasu hakchina loukhiba masi yamna phei*
Predicted	মসি য়াম্না ফৈ হায়না চীফ মিনিস্তর এন বীরেননসু হকচিন্না মরম ওইরগা মীনুংশি হংবা দেসাম *masi yamna phei haina chief minister n birennasu hakchinna maram oiraga minungshi hungba desam*
	Manipuri → English
Input	ডেসামনা পুখৎলকখিবা ৱাফম চীফ মিনিষ্টর এন বিরেননা হকচিন্না লৌখিবা অসিমক অফবা ওই *desamna pukhatlakkhiba wapham chief minister n birenna hakchinna loukhiba asimak aphaba oi*
Reference-1	it is good that the chief minister n biren took note of concerns raised by desam
Reference-2	it is good that chief minister n biren has taken due note of the voice raised by desam
Predicted	atsum raised the issue with the chief minister n biren took note of it and it is good

Table 7.8: Proposed model N-gram precisions with BLEU scores.

	Mni → En					*En → Mni*				
	BLEU	P_1	P_2	P_3	P_4	BLEU	P_1	P_2	P_3	P_4
Proposed Model	9.70	39.7	12.9	5.9	2.9	8.48	30.8	10.7	5.4	2.9

dataset settings and language pairs is necessary.

We further perform an error analysis to investigate the strengths and weaknesses of our best proposed MT setup (Monoses + Segmentation + Transliterations + Temporal Alignments). It is observed that the proposed model is capable of accurately generating unigram translations. Similarly, multi-word entities and two-gram translations are also correctly predicted in most cases. Some examples for both the translation directions are shown in Table 7.6. Matching colored texts represent correctly predicted unigram and multi-gram translations. However, the models frequently fail to handle higher multi-gram translations, resulting in a low overall BLEU scores. Table 7.8 shows the difference in BLEU score and the corresponding modified n-gram precisions P_n (n = *1,2,3,4*) for the model. With increase in n, the n-gram precision scores decrease significantly. We believe that the

difference in word order between the languages is a major contributor to the large disparity between the BLEU and n-gram precisions. In contrast to the Subject-Verb-Object (SVO) order in English, Manipuri follows the SOV order [214]. As a result, the unsupervised model fails to account for differences in word order. For instance, in the *En→Mni* translation example shown in Table 7.7, the order of the corresponding translation of *desam* (দেসাম) and *it is good* (মসি য়াম্না ꯑꯐꯩ) is reversed and is incorrectly predicted. Similar observations can also be seen for *Mni→En* MT.

We notice that presence of out-of-vocabulary terms (e.g., *metric*, সুপরিন্টেন্দেন্ট, etc. shown in Table 7.6) are still a major concert. This is mainly because the USMT models operate with a fixed vocabulary due to vocabulary cut-off during the generations of CLEs [11]. Spelling variation is another challenge that is unique to Manipuri text. The same word can have multiple spellings. In our corpus, for example, we discovered two different correct spellings (ডেসাম, দেসাম) of the same word *desam*. We believe that this issue of spelling variation contributes to data sparseness and that normalizing such variations would further improve translation performance.

The ChrF++ scores of above 30 points for both the MT directions indicate that the predicted translations of our our proposed model are reasonably accurate. One can get a reasonable understanding of the original text even though they are not grammatically perfect. Interestingly, there are also cases where the translated output is wrong, but it carries a similar semantic meaning to the reference. For instance, as shown in the example of *En → Mni* MT in Table 7.7, the word *desam* is wrongly translated as *atsum*. Both entities are names of two student unions in Manipur. This suggests that a temporal-aligned comparable corpus is a

viable option for cross-lingual embedding. It is also evident from the P_1 scores in Table 7.8 that the translation performance can be improved further by utilizing post-processing correction methods like neural language modelling [138], NMT hybridization [13, 144]. However, post-correction is not included within the scope of this study.

7.7 Summary

This chapter proposes a novel method for learning cross-lingual embeddings conditioned on comparable corpus temporal alignments. The resulting temporal embeddings can take advantage of translation features available across the alignments to capture more robust cross-lingual semantics relatedness. The proposed method first learns the sets of time-specific cross-lingual embeddings separately under the different temporal aligned conditions. They are then used to enhance the global cross-lingual embeddings by mapping them into common space using appropriate transformations. In total, we obtain significant improvements of about 70.62% and 59.80% in terms of BLEU score for the *En*→*Mni* and *Mni*→*En* MT, respectively, over the previous best unsupervised model on the language pair. Though not with high performance, this work provides a stable MT baseline for future research for the low-resource Manipuri-English language pair. We also performed an extensive qualitative analysis of the proposed model and offered several directions for future studies. Although we have exploited the month of the publication, every single date contains different articles in English and Manipuri versions in newspapers. We would also like to take advantage of such date-aligned features in the future.

8

Conclusion and Future Work

State-of-the-art machine translation models trained on a large parallel corpus have been reported to achieve excellent results, even comparable with human translations. On the other hand, millions of sentence pairs are normally necessary to develop a sound quality translation system. Unfortunately, most parallel corpora are limited for most language pairs, with few or no sentence pairs accessible. As a result, there is a massive gap between the advancement in translation technologies between low-resource and high-resource language pairs.

This thesis focuses on developing an MT system for low-resource Manipuri-English language pair, in which bilingual corpus is almost close to non-existence. As lack of technology inclusion would worsen the progress in processing capabilities, it may also push speakers of low-resource languages and dialects to high-resource languages with more substantial technical assistance. Therefore, it becomes critical to find technological solutions that compensate for resource constraints. This study contributes to this important problem by advancing the effort to develop an MT system for the low-resource Manipuri-English pair. Specifically, this thesis improves translation quality between the language pair without using expensive parallel sentences. The study also emphasizes minimum usage of language-specific resources so that the proposed techniques can be easily extended to other low-resource languages. Though not with high performance, this work provides a stable MT baseline for the low-resource Manipuri-English language pair. In the subsequent section, we summarize the contributions made in this thesis work. We further discuss the limitations of the proposed methods and possible future directions to explore.

8.1 Summary of Contributions

The contributions in this thesis work can broadly be divided into three categories. The initial group of contributions is concerned with the creation of the dataset. This thesis proposes a news domain Manipuri-English comparable corpus feasible for MT. We also develop tools to convert non-unicode Manipuri text to unicode as part of our corpus construction effort. Using manual and semi-automated procedures, the corpus is further aligned at the date and document levels. Furthermore,

the study also builds an MT evaluation dataset consisting of 1006 sentence pairs with two reference translations for each source sentence.

As the second contribution, this thesis develops two essential tools required for improving Manipuri-English MT. The first tool is a Manipuri suffix segmenter for normalizing the morphological inflection issue of Manipuri. From various experimental results, it is observed that segmenting the text significant improves the translation performance. Secondly, this thesis presents a transliteration model for transliterating English loanwords and named-entities to Manipuri. Specifically, we offer a neural hybrid machine transliteration model. Individual grapheme and phoneme-based models have limits, and the hybrid model overcomes them by simultaneously capturing grapheme and phoneme representations' properties.

The last group of contributions is concerned with the technological advancements made to the unsupervised SMT model. We make three significant contributions which are summarized below. Firstly, we propose two techniques for incorporating transliteration elements (produced using transliteration models) that will allow the USMT to establish a connecting link by exploiting phonetically similar words (transliteration pairs) between English and Manipuri: (1) Improving cross-lingual embeddings by exploiting automatically generated transliteration pairs. (2) Improving phrase-table using transliteration models. Experimental results show that the proposed methods can exploit the phonetically similar transliterated words between the language pair and further enhance translation performance. Secondly, this thesis proposes a method for exploiting document-level alignments to improve Manipuri-English MT performance. The study introduces a scoring module in USMT architecture, enabling the model to incorporate multiple translation features. Finally, we propose a novel method to incorporate tem-

porally aligned comparable document characteristics to enhance the translation performance by improving the cross-lingual embeddings. The proposed method first learns the sets of time-specific CLEs separately under the different temporal aligned conditions. They are then used to enhance the global CLEs by mapping them into common space via appropriate transformations.

8.2 Limitations and Future Works

This section highlights the limitations of the current study and some potential future research directions for the language pair MT.

1. **Datasets** : Large-sized datasets are necessary for various natural language processing task [49]. The dataset compiled in this study consists of 5.62M and 5.79M Manipuri and English tokens, respectively. Although the corpus is one of the largest available corpora for the language pair, it is still not at the same level as some other language pairs with more advanced language processing support. In the future, efforts may be directed toward collecting more data for the language pair. In particular, the Unicode conversion and the crawling procedure developed in this thesis, presented in Chapter 3, can be utilized to collect recently published articles from the Sangai Express and the Poknapham.

2. **Morphological Inflection Issue** : As already discussed in Section 5.7.1, we found that morphological inflection of Manipuri is a major reason for obtaining relatively lower performance of $En \rightarrow Mni$ compared to $Mni \rightarrow En$ for the BDI task. A similar problem was also reported in the earlier study [223] for other agglutinative languages like Estonian, Finnish, etc. Morphologi-

cal inflection leads to data sparsity which is a significant issue for models exploiting co-occurrence features. Research on developing an effective Manipuri segmenter to normalize these inflections is left as potential future work.

3. **The pre-ordering and post-ordering correction** : In Chapter 7, we have shown that the proposed MT models fail to handle word-order differences between Manipuri and English. It remains one of the major concerns affecting translation performance. This is a legit problem even for supervised SMT and NMT [22]. Several methods have been proposed to alleviate the issue comprising pre-ordering [104] and post-ordering [96] techniques. However, to the best of our knowledge, there has not been any attempt to incorporate reordering approaches on unsupervised MT. It is natural the problem will become more complicated in the unsupervised setting. Tackling this is an interesting problem.

4. **Towards Multi-lingual MT** : In recent years, research on multi-lingual MT has become immensely active and exciting. Several previous works have demonstrated that the low-resource language pairs benefit from following multi-lingual approaches [253, 98, 69]. Since English is utilized as a pivot language for most previous multi-lingual studies [8, 95], exploring this research direction for the Manipuri-English language pair may be a fruitful option.

A

Unicode Mapping Table

Tables A.1 and A.2 shows the complete mapping tables for converting the ASCII-based character(s) to the corresponding Unicode character(s) for Sangai Express and Poknapham Manipuri texts respectively.

Table A.1: Mapping Table for the Sangai Express texts. Integers inside the bracket represent the corresponding code points.

ASCII	Unicode	ASCII	Unicode	ASCII	Unicode
Òü (210 252)	ই (2439)	laü (108 161 252)	উ (2441)	Bą (66 161)	ক্ত (2453 2509 2453)
E (69)	ক্র (2453 2509 2545)	Aą (65 161)	ক (2453)	A (65)	ক্ (2453 2509)
ÑH (209 72)	স্ত (2488 2509 2453)	{ (123)	ি (2495)	Êą (202 161)	ষ্ট (2487 2509 2463)
ü (252)		þ (254)	ক (2453)	(8226)	ন (2472)
I (73)	ক্ষ (2453 2509 2480)	B (66)	ক্ত (2453 2509 2453)	T (84)	ঝ্ (2457 2509 2454)
j (106)	ট্ট (2463 2509 2463)	yûą (121 251 161)	ক্ষ (2453 2509 2480)	ą (339 161)	গু (2474 2509 2468)
Acą (65 162 161)	র্ক (2480 2509 2453)	G (71)	ক্ষ (2453 2509 2488)	âń (226 171)	ত্ব (2468 2509 2476)
x (120)	থ (2468 2509 2469)	Đ (208)	স্ট (2488 2509 2463)	ij (188)	ল্ল (2482 2509 2455)
r (114)	গু (2467 2509 2465)	û (251)	ণ (2467)	Â (194)	ল্ (2482 2509)
ê (234)	ু (2498)	(8221)	ন্ (2472 2509)	$ (36)	ঊ (2442)
‘ (96)	জ্ঞ (2460 2509 2462)	u (117)	ঋ (2468 2509 2478)	] (93)	
U (85)	ঙ্গ (2457 2509 2455)	k (107)	ঠ (2464)	ö (246)	র (2480)
ń (171)	ূ (2509 2476)	z (8221 122)	ন্ত (2472 2509 2468)	ű (182)	ম (2478)
ÿ (184)	ৗ (2509 2479)	æ (230)	ু (2497)	a (97)	জ্ব (2460 2509 2476)
(8221 8218)	ন্থ (2472 2509 2469)	H (72)		(8218)	থ (2469)
(8216)	” (34)	(8217)	” (34)	Q (81)	ঘ (2456)
õ (245)	ূ (2499)	S (83)	ঙ্ক (2457 2509 2453)	đÃ (164 195)	ব্র (2476 2509 2482)
Çą (199 161)	শু (2486 2497)	vąûą (118 161 251 161)	ক্ত (2453 2509 2468)	şć (179 162)	র্ম (2480 2509 2478)
(8249)	ধ (2471)	Ûą (219 161)	ক্ষ (2453 2509 2487)	vąû (118 161 251)	ক্ত (2453 2509 2468)
b (98)	জ্ঞ (2460 2509 2480)	¿ (190)	ল্ব (2482 2509 2476)	Á (193)	ল্ড (2482 2509 2465)
F (70)	ঋ (2453 2509 2478)	ò (242)	ঁ (2433)	+ (43)	ঔ (2452)
f (102)	ন্স (2472 2509 2488)	£ (191)	ল্ (2482 2509)	(710)	ন্ম (2470 2509 2478)
Ç (199)	শু (2486 2497)	ŧ (180 353)	ষ্প (2478 2509 2474)	(8240)	দ্র (2470 2509 2480)
ż (187)	ঙ্ক (2482 2509 2453)	Áą (193 161)	ল্ড (2482 2509 2465)	Ñ| (209 124)	স্ত্র (2488 2509 2468 2509 2480)
(8482)	য (2479)	(376)	প্ (2474 2509)	ÒÇü (210 199 252)	ইশু (2439 2486 2497)
ð (240)	জ্জ (2460 2509 2460)	(8222)	দ্দ (2470 2509 2470)	SS (223)	প্র (2474 2509 2480)
y (121)	ত্র (2468 2509 2480)	â (226)	ত্ (2468 2509)	L (76)	ল্ল (2455 2509 2455)
(8220)	ভ (2472 2509 2465)	Ñ (209)	স্ (2488 2509)	Ñ (209 353)	ষ্প (2488 2509 2474)
e (101)	ঞ্চ (2462 2509 2458)	v (118)	ত্ত (2468 2509 2468)	Ø (216)	ড় (2524)
W (87)	চ (2458)	á (225)	ছ (2459)	# (35)	ঈ (2440)
% (37)	% (37)	z (122)	ও (2451)	Ò (213)	ষ্ম (2489 2509 2478)
ô (244)			(124)	ý (253)	ধ (2471)
Þ (222)	ন্ (2472 2509)	V (86)	ঙ্ (2457 2509)	/ (47)	ব্ (2476 2509)
Wą (87 161)	চ (2458)	Rą (82 161)	ঙ (2457)	ią (105 161)	ট (2463)
i (105)	ট (2463)	tą (116 161)	ত (2468)	Cą (67 161)	ষ্ট (2453 2509 2463)
vą (118 161)	ত্ত (2468 2509 2468)	ø (248)	ৃ (2509 2480)	Ì (204)	ঞ্চ (2487 2509 2467)
C (67)	ষ্ট (2453 2509 2463)	ç (231)	ূ (2497)	c (99)	ঝ (2461)
Z (90)	চ্ (2458 2509)	ß (255)		œ (247)	র (2480)
(8225)	দ্ব (2470 2509 2476)	d (100)	ঞ (2462)	čą (163 161)	ফ (2475)
č (163)	ফ (2475)	óą (243 161)	ফ (2475)	ó (243)	ফ (2475)
çą (231 161)	ূ (2497)	óøą (243 248 161)	ফ্ (2475 2509 2480)	N (78)	গ (2455)
ý (8225 253)	দ্ধ (2470 2509 2471)	" (34)	অ (2437)	à (224)	া (2494)
[(91)	ি (2495)	ã (227)	ী (2496)	å (229)	ূ (2497)
èą (232 161)	ূ (2498)	è (232)	ূ (2498)	Ò (212)	ই (2489 2509 2476)
n (110)	ট (2466)	ampersand (38)	এ (2447)	ë (235)	ে (2503)
ì (236)	ে (2503)	'' (34 34)	ঐ (2448)	í (237)	ৈ (2504)
î (238)	ৈ (2504)	* (42)	ও (2451)	; (59)	ৎ (2510)
} (125)	ং (2434)	J (74)	খ (2454)	K (75)	গ (2455)
t (116)	ত (2468)	Æţ (198 181)	শ্ম (2486 2509 2478)	ţ (181)	ম (2478)
ć (162)	র্ (2480 2509)	ćą (162 161)	র্ (2480 2509)	ŕ (175)	র (2545)
ą (161)		o (111)	ণ (2467)	> (62)	ন (2472)
= (61)	থ (2469)	> (62)	ন (2472)	g (103)	জ্ঞ (2462 2509 2460)
Þê (222 234)	দ্ধ (2472 2509 2471)	é (233)		(339)	গু (2474 2509 2468)
ñ (241)	ও (2451)	(353)	প (2474)	Ù (217)	প্ল (2474 2509 2474)
đ (164)	ব (2476)	ň (172)	ব (2476)	ő (174)	ভ (2477)
Øn (216 110)	ঢ় (2525)	(8250)	ন্স (2474 2509 2488)	ę (166)	ব্দ (2476 2509 2470)
(732)	ঋ (2443)	ş (179)	ম (2478)	ť (180)	ম্ (2478 2509)
Ú (218)	য় (2527)	ź (185)	র (2480)	Å (197)	শ (2486)
R (82)	ঙ (2457)	Ý (221)	ক্ষ (2453 2509 2487 2509 2467)	Æ (198)	শ (2486)
Î (206)	স (2488)	Ê (202)	স্ (2488 2509)	Ò (210)	হ (2489)
@ (64)	ঃ (2435)	È (200)	ষ (2487)	(402)	দ (2470)
ž (186)	ল (2482)	\\ (92 92)	জ (2460)	É (201)	ধ (2487 2509 2476)
À (192)	ল্ল (2482 2509 2482)	Ăą (195 161)	ল (2482)	Ā (195)	ল (2482)
(8211)	ন্ (2472 2509)	ą (8212 161)	ন (2472)	(8212)	ন (2472)
Ä (196)	ম্ (2472 2509 2472)	X (88)	ন্স (2472 2509 2488)	P (80)	গু (2455 2497)
Œ (215)	হ (2489 2497)	M (77)	গ্ব (2455 2509 2476)	Ë (203)	ঠ (2487 2509 2464)
Ü (220)	ক্ষ্ম (2453 2509 2487 2509 2478)	sn (115 110)	ণ্ড (2467 2509 2466)	ś (177)	স্ত (2478 2509 2477)
Ó (211)	হ্ল (2489 2509 2482)	ìà (236 224)	ো (2507)	ëà (235 224)	ো (2507)
ïĭ (236 239)	ৌ (2508)	ëï (235 239)	ৌ (2508)	l (108)	ড (2465)
lą (108 161)	ড (2465)	Ølą (216 108 161)	ড় (2524)	0 (48)	০ (2534)
1 (49)	১ (2535)	2 (50)	২ (2536)	3 (51)	৩ (2537)
4 (52)	৪ (2538)	5 (53)	৫ (2539)	6 (54)	৬ (2540)
7 (55)	৭ (2541)	8 (56)	৮ (2542)	9 (57)	৯ (2543)
ú (250)	। (2404)	'à (34 224)	আ (2438)	vòą (118 246 161)	ত্র (2468 2509 2468 2509 2480)
đö (164 246)	ব্র (2476 2509 2480)	ş (179 8212)	ন্ম (2478 2509 2472)	ŒÿĬ (215 184)	হ্ম্য (2489 2509 2527 2497)
Aąą (65 161 8212 161)	ক্ক (2453 2509 2472)	Jąø (74 161 248)	ঋ (2454 2509 2480)	Jń (74 171)	খ্র (2454 2509 2545)
gæÿ (103 230 184)	জ্ঞ্য (2460 2509 2479 2497)	Aâąÿ (65 229 161 184)	ক্ষ্য (2453 2509 2479 2497)	iąö (105 161 246)	ট্র (2463 2509 2480)
Êąö (202 161 246)	ষ্ট্র (2487 2509 2463 2509 2480)	ãÿ (229 184)	ু (2509 2479 2497)	âąÿ (229 161 184)	ু (2509 2479 2497)
Aâą (65 229 161)	কু (2453 2497)	(353 8212)	প্ল (2474 2509 2472)	yû (121 251)	ক্ত (2453 2509 2480)

Table A.2: Mapping Table for the Poknapham texts. Integers inside the bracket represent the corresponding code points.

ASCII	Unicode	ASCII	Unicode	ASCII	Unicode
Òü (210 252)	ই (2439)	laü (108 161 252)	উ (2441)	Bą (66 161)	ক্ত (2453 2509 2453)
E (69)	ক্র (2453 2509 2545)	Aą (65 161)	ক (2453)	A (65)	ক্ (2453 2509)
N̄H (209 72)	স্ক (2488 2509 2453)	{ (123)	ি (2495)	Êą (202 161)	ষ্ট (2487 2509 2463)
ü (252)		þ (254)	ক (2453)	(8226)	ন (2472)
I (73)	ক্ষ (2453 2509 2480)	B (66)	ক্ত (2453 2509 2453)	T (84)	ঋ (2457 2509 2454)
j (106)	ট্ট (2463 2509 2463)	yûą (121 251 161)	ক্ষ (2453 2509 2480)	ą (339 161)	গু (2474 2509 2468)
Aćą (65 162 161)	র্ক (2480 2509 2453)	Ać (65 162)	র্ক (2480 2509 2453)	G (71)	ক্স (2453 2509 2488)
âń (226 171)	ত্ত (2468 2509 2476)	x (120)	থ (2468 2509 2469)	Đ (208)	স্ট (2488 2509 2463)
ij (188)	ল্ল (2482 2509 2455)	r (114)	ও (2467 2509 2465)	û (251)	ণ (2467)
Ă (194)	ল্ (2482 2509)	ê (234)	ু (2498)	(8221)	ন্ (2472 2509)
$ (36)	উ (2442)	‘ (96)	জ্ঞ (2460 2509 2462)	u (117)	ত্ম (2468 2509 2478)
] (93 32)		U (85)	ঙ্গ (2457 2509 2455)	k (107)	ঠ (2464)
ö (246)	র (2480)	ń (171)	় (2509 2476)	z (8221 122)	ন্ত (2472 2509 2468)
ű (182)	ম (2478)	ÿ (184)	◌ঽ (2509 2479)	æ (230)	ু (2497)
a (97)	জ্ঞ (2460 2509 2476)	(8221 8218)	ন্থ (2472 2509 2469)	H (72)	
(8218)	থ (2469)	(8216)	” (34)	(8217)	” (34)
Q (81)	ঘ (2456)	õ (245)	ূ (2499)	S (83)	ঙ্ক (2457 2509 2453)
đĂ (164 195)	ব্র (2476 2509 2482)	Çą (199 161)	শু (2486 2497)	vąûą (118 161 251 161)	ক্ত (2453 2509 2468)
şć (179 162)	র্ম (2480 2509 2478)	(8249)	ধ (2471)	Ũą (219 161)	ক্ষ (2453 2509 2487)
vąû (118 161 251)	ক্ত (2453 2509 2468)	b (98)	জ্ঞ (2460 2509 2480)	Ũ (219)	ক্ষ (2453 2509 2487)
¿ (190)	ন্ধ (2482 2509 2476)	Á (193)	ল্ড (2482 2509 2465)	F (70)	ক্ম (2453 2509 2478)
ò (242)	ঁ (2433)	+ (43)	ঔ (2452)	f (102)	ঙ্গ (2472 2509 2488)
£ (191)	ল্ (2482 2509)	(710)	দ্ম (2470 2509 2478)	Ç (199)	শু (2486 2497)
ŧ (180 353)	ম্প (2478 2509 2474)	(8240)	ব্র (2470 2509 2480)	ż (187)	ল্ক (2482 2509 2453)
Áą (193 161)	ল্ড (2482 2509 2465)	N̄ (209 124)	স্ত্র (2488 2509 2468 2509 2480)	(8482)	ম (2479)
(376)	প্ (2474 2509)	ÒÇü (210 199 252)	ইশু (2439 2486 2497)	ią (8211 105 161)	ন্ট (2472 2509 2463)
ð (240)	জ্জ (2460 2509 2460)	(8222)	দ্দ (2470 2509 2470)	SS (223)	প্র (2474 2509 2480)
y (121)	ব্র (2468 2509 2480)	ã (226)	ত্ (2468 2509)	L (76)	গ্ল (2455 2509 2455)
(8220)	ন্ড (2472 2509 2465)	Ñ (209)	স্ (2488 2509)	N (209 353)	স্প (2488 2509 2474)
e (101)	ষ্ক (2462 2509 2458)	v (118)	ত্ত (2468 2509 2468)	Ø (216)	ড় (2524)
W (87)	চ (2458)	á (225)	ছ (2459)	# (35)	ঈ (2440)
% (37)	% (37)	z (122)	ও (2451)	Ó (213)	ক্ষ (2453 2509 2487)
õ (244)		| (124)	ব্র (2468 2509 2480)	ý (253)	ধ (2471)
Þ (222)	ন্ (2472 2509)	V (86)	ঙ্ (2457 2509)	/ (47)	ব্ (2476 2509)
ŋ (173)	Wą (87 161)	চ (2458)	Rą (82 161)	ta (116 161)	ঙ (2457)
ią (105 161)	ট (2463)	i (105)	ট (2463)	tą (116 161)	ত (2468)
Cą (67 161)	ক্ট (2453 2509 2463)	va (118 161)	ত্ত (2468 2509 2468)	ø (248)	◌ু (2509 2480)
Î (204)	ষ্ক (2487 2509 2467)	C (67)	ক্ট (2453 2509 2463)	ç (231)	ু (2497)
c (99)	ঝ (2461)	Z (90)	চ্ (2458 2509)	ß (255)	
œ (247)	র (2480)	(8225)	দ্ধ (2470 2509 2476)	d (100)	ঞ (2462)
čą (163 161)	ফ (2475)	č (163)	ফ (2475)	óą (243 161)	ফ (2475)
ó (243)	ফ (2475)	çą (231 161)	ু (2497)	óóą (243 248 161)	ফ্ল (2475 2509 2480)
N (78)	গ (2455)	ý (8225 253)	দ্ধ (2470 2509 2471)	" (34)	অ (2437)
à (224)	◌া (2494)	[(91)	ি (2495)	ã (227)	ী (2496)
ă (229)	ু (2497)	èą (232 161)	ু (2498)	è (232)	ু (2498)
Ò (212)	হ্ (2489 2509 2476)	n (110)	ঢ (2466)	& (38)	এ (2447)
ë (235)	◌ো (2503)	ì (236)	◌ো (2503)	'' (34 34)	ঐ (2448)
í (237)	ৈ (2504)	î (238)	ৈ (2504)	* (42)	ও (2451)
; (59)	ৎ (2510)	} (125)	◌ং (2434)	J (74)	খ (2454)
K (75)	গ (2455)	t (116)	ত (2468)	Æţ (198 181)	শ্ম (2486 2509 2478)
ţ (181)	ম (2478)	ć (162)	র্ (2480 2509)	ćą (162 161)	র্ (2480 2509)
ŕ (175)	র (2545)	ą (161)		o (111)	ণ (2467)
> (62)	ন (2472)	= (61)	থ (2469)	> (62)	ন (2472)
g (103)	ঞ্জ (2462 2509 2460)	Þê (222 234)	ন্ধ (2472 2509 2471)	é (233)	
(339)	গু (2474 2509 2468)	ñ (241)	ও (2451)	(353)	প (2474)
Ŭ (217)	গ্ল (2474 2509 2474)	d (164)	ব (2476)	ň (172)	ব (2476)
ő (174)	ভ (2477)	Ón (216 110)	ঢ় (2525)	(8250)	ঙ্গ (2474 2509 2488)
ę (166)	ব্দ (2476 2509 2470)	(732)	ঋ (2443)	ş (179)	ম (2478)
ŧ (180)	ম্ (2478 2509)	Ú (218)	য় (2527)	ź (185)	র (2480)
Å (197)	শ (2486)	R (82)	ঙ (2457)	Ý (221)	ক্ষ (2453 2509 2487 2509 2467)
Æ (198)	শ (2486)	Î (206)	স (2488)	É (202)	স্ (2488 2509)
Ò (210)	হ (2489)	È (200)	ষ (2487)	(402)	দ (2470)
ž (186)	ল (2482)	\\ (92 92)	জ (2460)	É (201)	ধ (2487 2509 2476)
À (192)	ল্ল (2482 2509 2482)	Ăą (195 161)	ল (2482)	Ã (195)	ল (2482)
ą (8212 161)	ন (2472)	Ä (196)	ম্ন (2472 2509 2472)	X (88)	ঙ্গ (2472 2509 2488)
P (80)	গু (2455 2497)	Œ (215)	ধ (2489 2497)	M (77)	গ্ধ (2455 2509 2476)
É (203)	ষ্ঠ (2487 2509 2464)	Û (220)	ক্ষ্ম (2453 2509 2487 2509 2478)	sn (115 110)	ণ্ড (2467 2509 2466)
ś (177)	স্ত (2478 2509 2477)	Ó (211)	হ্ল (2489 2509 2482)	ìà (236 224)	◌ো (2507)
ëà (235 224)	◌ো (2507)	ïï (236 239)	◌ৌ (2508)	ëï (235 239)	◌ৌ (2508)
l (108)	ঙ (2465)	lą (108 161)	ঙ (2465)	Ól (216 108)	ড় (2524)
0 (48)	০ (2534)	1 (49)	১ (2535)	2 (50)	২ (2536)
3 (51)	৩ (2537)	4 (52)	৪ (2538)	5 (53)	৫ (2539)
6 (54)	৬ (2540)	7 (55)	৭ (2541)	8 (56)	৮ (2542)
9 (57)	৯ (2543)	ú (250)	। (32 2404)	- (45)	- (45)
((40)	((40)	) (41)	) (41)	. (46)	. (46)
? (63)	? (63)	! (33)	! (33)	ěą (165 161)	ন (2472)
(8212)	◌ঃ (2435)	(8211)	◌ঃ (2435)	@ (64)	ন্ (2472 2509)
ě (353 165)	গ্ল (2474 2509 2472)	Nè (78 165)	গ্ল (2455 2509 2472)	şé (179 165)	ম্ন (2478 2509 2472)
âě (226 165)	ত্র (2468 2509 2472)	ř (176)	ভ (2477 2509 2480)	ě (165)	ন (2472)
(338)	ব (2476)	(956)	ম (2478)	đõ (164 246)	ব্র (2476 2509 2480)
Aå (65 229)	কু (2453 2497)	Òů (210 251)	হ্ন (2489 2509 2472)	åå (229 229)	ু (2497)
Ă (353 195)	গ্ল (2474 2509 2482)	Aè (65 232)	কু (2453 2498)	°à (34 224)	আ (2438)
Ňz (209 122)	স্ত (2488 2509 2468)	Aåaў (65 229 161 184)	ক্যু (2453 2509 2479 2497)	yû (121 251)	ক্ষ (2453 2509 2480)
åÿ (229 184)	◌ৃ (2509 2479 2497)	N̄Þ (209 72 254)	স্ক (2488 2509 2453)	Ăń (197 172)	শ্র (2486 2509 2545)
Œÿ (215 184)	হ্যু (2489 2509 2527 2497)	Jń (74 171)	খ্র (2454 2509 2545)	gæў (103 230 184)	জ্যু (2460 2509 2479 2497)
q (113)	ষ্ঠ (2472 2509 2464)				

B

Machine Translation Evaluation Matrices

B.1 BLEU

BiLingual Evaluation Understudy or BLEU is a metric for automatically evaluating MT outputs [168]. Generally, BLEU score is represented as a number between 0 and 100. It measures the similarity of the machine-translated outputs with a set of high quality reference translations. A value of 0 means that the machine-translated candidate output has no overlap with the reference translation, while a

value of 100 means there is perfect overlap with the reference translations. Mathematically, BLEU is define as follows:

$$BLEU = min(1, exp(1 - \frac{reference\ length}{candidate\ length}))(\prod_{i=1}^{n} P_i)^{\frac{1}{n}} \times 100 \tag{B.1}$$

where, length are computed in terms of n-grams, and P_i is the modified precision for n-gram. Typically, BLEU considers n = 4.

B.2 ChrF++

ChrF++ [171] is an extension of ChrF (Character n-gram F-score) [170] by adding word n-grams. It is based on n-gram based F-scores defined as follows:

$$ngrF\beta = (i + \beta^2))\frac{ngrP \cdot ngrR}{\beta^2 \cdot ngrP + ngrR} \tag{B.2}$$

where β is a parameter. $ngrP$ and $ngrR$ are n-gram precision and recall averaged arithmetically over all n-grams. ChrF++ score is achieve by combining the word n-grams with the character n-grams and averaging together.

References

[1] Abbas, M. R. & Asif, D. K. H. (2020). Punjabi to iso 15919 and roman transliteration with phonetic rectification. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(2), 1–20.

[2] Adafre, S. F. & De Rijke, M. (2006). Finding similar sentences across multiple languages in wikipedia. In *Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources*.

[3] Aker, A., Kanoulas, E., & Gaizauskas, R. J. (2012). A light way to collect comparable corpora from the web. In *Proceedings of LREC 2012* (pp. 15–20).: Citeseer.

[4] Al-Onaizan, Y. & Knight, K. (2002a). Machine transliteration of names in arabic text. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages* (pp. 1–13).: Association for Computational Linguistics.

[5] Al-Onaizan, Y. & Knight, K. (2002b). Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 400–408).: Association for Computational Linguistics.

[6] Alam, M. & Hussain, S. u. (2020). Deep learning based roman-urdu to urdu transliteration. *International Journal of Pattern Recognition and Artificial Intelligence*.

[7] Ammar, W., Mulcaire, G., Tsvetkov, Y., Lample, G., Dyer, C., & Smith, N. A. (2016). Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

[8] Arivazhagan, N., Bapna, A., Firat, O., Aharoni, R., Johnson, M., & Macherey, W. (2019). The missing ingredient in zero-shot neural machine translation. *arXiv preprint arXiv:1903.07091*.

[9] Armentano-Oller, C., Carrasco, R. C., Corbi-Bellot, A. M., Forcada, M. L., Ginesti-Rosell, M., Ortiz-Rojas, S., Perez-Ortiz, J. A., Ramirez-Sanchez, G., Sanchez-Martinez, F., & Scalco, M. A. (2006). Open-source portuguese-spanish machine translation. In *PROPOR* (pp. 50–59).: Springer.

[10] Artetxe, M., Labaka, G., & Agirre, E. (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[11] Artetxe, M., Labaka, G., & Agirre, E. (2018b). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 789–798).

[12] Artetxe, M., Labaka, G., & Agirre, E. (2018c). Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* Brussels, Belgium: Association for Computational Linguistics.

[13] Artetxe, M., Labaka, G., & Agirre, E. (2019). An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 194–203).

[14] Artetxe, M., Labaka, G., Agirre, E., & Cho, K. (2018d). Unsupervised neural machine translation. In *International Conference on Learning Representations*.

[15] Artetxe, M. & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7, 597–610.

[16] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[17] Banerjee, T., Kunchukuttan, A., & Bhattacharyya, P. (2018). Multilingual indian language translation system at wat 2018: Many-to-one phrase-based smt. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*.

[18] Bansal, A., Banerjee, E., & Jha, G. N. (2013). Corpora creation for indian language technologies–the ilci project. In *the sixth Proceedings of Language Technology Conference (LTC '13)*.

[19] Barros, M. J. & Weiss, C. (2006). Maximum entropy motivated grapheme-to-phoneme, stress and syllable boundary prediction for portuguese text-to-speech. *IV Jornadas en Tecnologías del Habla. Zaragoza, Spain*, (pp. 177–182).

[20] Bilac, S. & Tanaka, H. (2005). Direct combination of spelling and pronunciation information for robust back-transliteration. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 413–424).: Springer.

[21] Bisani, M. & Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5), 434–451.

[22] Bisazza, A. & Federico, M. (2016). A survey of word reordering in statistical machine translation: Computational models and language phenomena. *Computational linguistics*, 42(2), 163–205.

[23] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3, 993–1022.

[24] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.

[25] Bouamor, H. & Sajjad, H. (2018). H2@ bucc18: Parallel sentence extraction from comparable corpora using multilingual sentence embeddings. In *Proc. Workshop on Building and Using Comparable Corpora*.

[26] Britz, D., Goldie, A., Luong, T., & Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

[27] Buck, C. & Koehn, P. (2016). Quick and reliable document alignment via tf/idf-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers* (pp. 672–678).

[28] Chaudhury, S., Sen, S., & Nandi, G. R. (2012). A finite state transducer (fst) based font converter. *International Journal of Computer Applications*, 58(17), 35–39.

[29] Chelliah, S. L. (1990). Level-ordered morphology and phonology in manipuri. *Linguistics of the Tibeto-Burman Area*, 13(2), 27–72.

[30] Chen, N., Banchs, R. E., Zhang, M., Duan, X., & Li, H. (2018). Report of news 2018 named entity transliteration shared task. In *Proceedings of the Seventh Named Entities Workshop* (pp. 55–73).

[31] Chen, P., Bogoychev, N., Heafield, K., & Kirefu, F. (2020). Parallel sentence mining by constrained decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1672–1678).

[32] Chen, Y., Liu, Y., Cheng, Y., & Li, V. O. (2017). A teacher-student framework for zero-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1925–1935).

[33] Cheng, Y. (2019). Joint training for pivot-based neural machine translation. In *Joint Training for Neural Machine Translation* (pp. 41–54). Springer.

[34] Cheon, J. & Ko, Y. (2021). Parallel sentence extraction to improve cross-language information retrieval from wikipedia. *Journal of Information Science*, 47(2), 281–293.

[35] Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 263–270).: Association for Computational Linguistics.

[36] Chitwirat, P., Facundes, N., & Sirinaovakul, B. (2008). English-thai machine translation in a lexicalist grammar. In *Communications and Information Technologies, 2008. ISCIT 2008. International Symposium on* (pp. 171–174).: IEEE.

[37] Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014a). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[38] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[39] Choudhary, N. & Jha, G. N. (2011). Creating multilingual parallel corpora in indian languages. In *Language and Technology Conference* (pp. 527–537).: Springer.

[40] Choudhury, S. I., Singh, L. S., Borgohain, S., & Das, P. K. (2004). Morphological analyzer for manipuri: Design and implementation. In *Asian Applied Computing Conference* (pp. 123–129).: Springer.

[41] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[42] Conneau, A. & Lample, G. (2019). Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems* (pp. 7059–7069).

[43] Conneau, A., Lample, G., Ranzato, M., Denoyer, L., & Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

[44] Currey, A., Barone, A. V. M., & Heafield, K. (2017). Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation* (pp. 148–156).

[45] Dabre, R., Chu, C., & Kunchukuttan, A. (2020). A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5), 1–38.

[46] Dabre, R., Fujita, A., & Chu, C. (2019). Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 1410–1416).

[47] Dara, A. A. & Lin, Y.-C. (2016). Yoda system for wmt16 shared task: Bilingual document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers* (pp. 679–684).

[48] Dash, N. S. (2013). Linguistic divergences in english to bengali translation. *International Journal of English Linguistics*, 3(1), 31.

[49] Dash, N. S. & Chaudhuri, B. B. (2001). Why do we need to develop corpora in indian languages. In *the International Working Conference on Sharing Capability in Localisation and Human Language Technologies SCALLA-2001. Bangalore*: Citeseer.

[50] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[51] Dinu, G., Lazaridou, A., & Baroni, M. (2014). Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

[52] Divay, M. & Vitale, A. J. (1997). Algorithms for grapheme-phoneme translation for english and french: Applications for database searches and speech synthesis. *Computational linguistics*, 23(4), 495–523.

[53] Dorr, B. J. (1993). Interlingual machine translation a parameterized approach. *Artificial Intelligence*, 63(1), 429–492.

[54] Dou, Q., Vaswani, A., & Knight, K. (2014). Beyond parallel data: Joint word alignment and decipherment improves machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 557–565).

[55] Dowling, M., Lynn, T., Poncelas, A., & Way, A. (2018). Smt versus nmt: Preliminary comparisons for irish. In *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)* (pp. 12–20).

[56] Duan, S., Zhao, H., Zhang, D., & Wang, R. (2020a). Syntax-aware data augmentation for neural machine translation. *arXiv preprint arXiv:2004.14200*.

[57] Duan, X., Ji, B., Jia, H., Tan, M., Zhang, M., Chen, B., Luo, W., & Zhang, Y. (2020b). Bilingual dictionary based neural machine translation without using parallel sentences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1570–1579).

[58] Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 489–500).

[59] Ezen-Can, A. (2020). A comparison of lstm and bert for small corpus. *arXiv preprint arXiv:2009.05451*.

[60] Fadaee, M. & Monz, C. (2018). Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 436–446).

[61] Finch, A. & Sumita, E. (2010). Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model. In *Proceedings of the 2010 Named Entities Workshop* (pp. 48–52).: Association for Computational Linguistics.

[62] Firat, O., Sankaran, B., Al-Onaizan, Y., Vural, F. T. Y., & Cho, K. (2016). Zero-resource translation with multi-lingual neural machine translation. In

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 268–277).

[63] Glavaš, G., Litschko, R., Ruder, S., & Vulić, I. (2019). How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 710–721).

[64] Gomes, L. & Lopes, G. (2016). First steps towards coverage-based document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers* (pp. 697–702).

[65] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[66] Goyal, V., Kumar, A., & Lehal, M. S. (2020). Document alignment for generation of english-punjabi comparable corpora from wikipedia. *International Journal of E-Adoption (IJEA)*, 12(1), 42–51.

[67] Grundkiewicz, R. & Heafield, K. (2018). Neural machine translation techniques for named entity transliteration. In *Proceedings of the Seventh Named Entities Workshop* (pp. 89–94).

[68] Gu, J., Hassan, H., Devlin, J., & Li, V. O. (2018a). Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 344–354).

[69] Gu, J., Wang, Y., Chen, Y., Li, V. O., & Cho, K. (2018b). Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3622–3631).

[70] Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., & Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

[71] Guo, M., Shen, Q., Yang, Y., Ge, H., Cer, D., Abrego, G. H., Stevens, K., Constant, N., Sung, Y.-H., Strope, B., et al. (2018). Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers* (pp. 165–176).

[72] Guzmán, F., Chen, P.-J., Ott, M., Pino, J., Lample, G., Koehn, P., Chaudhary, V., & Ranzato, M. (2019). The flores evaluation datasets for low-resource machine translation: Nepali–english and sinhala–english. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 6100–6113).

[73] Gv, R. (1995). Python tutorial: Technical report cs-r9526. *Amsterdam: Centrum voor Wiskunde en Informatica*.

[74] Ha, T.-L., Niehues, J., & Waibel, A. (2018). Effective strategies in zero-shot neural machine translation. *Final Report on Under-Resourced Languages*.

[75] Haddow, B. & Kirefu, F. (2020). Pmindia–a collection of parallel corpora of languages of india. *arXiv preprint arXiv:2001.09907*.

[76] Haffari, G. & Sarkar, A. (2009). Active learning for multilingual statistical machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 181–189).

[77] Haizhou, L., Min, Z., & Jian, S. (2004). A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Annual Meeting on association for Computational Linguistics* (pp. 159).: Association for Computational Linguistics.

[78] Hangya, V., Braune, F., Kalasouskaya, Y., & Fraser, A. (2018). Unsupervised parallel sentence extraction from comparable corpora. In *International Workshop on Spoken Language Translation*.

[79] Hangya, V. & Fraser, A. (2019). Unsupervised parallel sentence extraction with parallel segment detection helps machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1224–1234).

[80] Hazem, A. & Morin, E. (2016). Efficient data selection for bilingual terminology extraction from comparable corpora. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 3401–3411).

[81] Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation* (pp. 187–197).: Association for Computational Linguistics.

[82] Heaps, H. S. (1978). *Information retrieval, computational and theoretical aspects*. Academic Press.

[83] Henríquez, C., Costa-jussá, M. R., Banchs, R. E., Formiga, L., & Mariño, J. B. (2011). Pivot strategies as an alternative for statistical machine translation tasks involving iberian languages. In *Workshop on ICL NLP Tasks* (pp. 22–27).

[84] Hermann, K. M. & Blunsom, P. (2014). Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 58–68).

[85] Hermjakob, U., Knight, K., & Daumé III, H. (2008). Name translation in statistical machine translation-learning when to transliterate. In *Proceedings of ACL-08: HLT* (pp. 389–397).

[86] Hill, F., Cho, K., & Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1367–1377).

[87] Hoang, H., Koehn, P., & Lopez, A. (2009). A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *nnnn*.

[88] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

[89] Hong, G., Kim, M.-J., Lee, D.-G., & Rim, H.-C. (2009). A hybrid approach to english-korean name transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration* (pp. 108–111).: Association for Computational Linguistics.

[90] Huang, D., Zhao, L., Li, L., & Yu, H. (2010). Mining large-scale comparable corpora from chinese-english news collections. In *Coling 2010: Posters* (pp. 472–480).

[91] Hutchins, W. J. & Somers, H. L. (1992). *An introduction to machine translation*, volume 362. Academic Press London.

[92] Imamura, K. & Sumita, E. (2018). Nict self-training approach to neural machine translation at nmt-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation* (pp. 110–115).

[93] Jean, S., Firat, O., Cho, K., Memisevic, R., & Bengio, Y. (2015). Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation* (pp. 134–140).

[94] Jha, G. N. (2012). The tdil program and the indian language corpora initiative. In *Language Resources and Evaluation Conference*.

[95] Ji, B., Zhang, Z., Duan, X., Zhang, M., Chen, B., & Luo, W. (2020). Cross-lingual pre-training based transfer for zero-shot neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34 (pp. 115–122).

[96] Jia, Y., Carl, M., & Wang, X. (2019). How does the post-editing of neural machine translation compare with from-scratch translation? a product and process study. *The Journal of Specialised Translation*, 31, 60–86.

[97] Jiampojamarn, S., Kondrak, G., & Sherif, T. (2007). Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference* (pp. 372–379).

[98] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5, 339–351.

[99] Jung, S. Y., Hong, S., & Paek, E. (2000). An english to korean transliteration model of extended markov window. In *Proceedings of the 18th conference on Computational linguistics-Volume 1* (pp. 383–389).: Association for Computational Linguistics.

[100] Kalchbrenner, N. & Blunsom, P. (2013). Recurrent continuous translation models. In *EMNLP*, volume 3 (pp. 413).

[101] Kang, B.-J. & Choi, K.-S. (2000). Automatic transliteration and back-transliteration by decision tree learning. In *LREC*: Citeseer.

[102] Karakanta, A., Dehdari, J., & van Genabith, J. (2018). Neural machine translation for low-resource languages without parallel corpora. *Machine Translation*, 32(1), 167–189.

[103] Karimi, S., Scholer, F., & Turpin, A. (2011). Machine transliteration survey. *ACM Computing Surveys (CSUR)*, 43(3), 17.

[104] Kawara, Y., Chu, C., & Arase, Y. (2018). Recursive neural network based preordering for english-to-japanese machine translation. In *Proceedings of ACL 2018, Student Research Workshop* (pp. 21–27).

[105] Kementchedjhieva, Y., Hartmann, M., & Søgaard, A. (2019). Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3336–3341).

[106] Keung, P., Salazar, J., Lu, Y., & Smith, N. A. (2020). Unsupervised bitext mining and translation via self-trained contextual embeddings. *Transactions of the Association for Computational Linguistics*, 8, 828–841.

[107] Kim, Y., Gao, Y., & Ney, H. (2019a). Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1246–1257).

[108] Kim, Y., Graça, M., & Ney, H. (2020). When and why is unsupervised neural machine translation useless? *arXiv preprint arXiv:2004.10581*.

[109] Kim, Y., Petrov, P., Petrushkov, P., Khadivi, S., & Ney, H. (2019b). Pivot-based transfer learning for neural machine translation between non-english languages. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 866–876).

[110] Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[111] Kishida, K. & Chen, K.-h. (2021). Experiments on cross-language information retrieval using comparable corpora of chinese, japanese, and korean languages. In *Evaluating Information Retrieval and Access Tasks* (pp. 21–37).: Springer, Singapore.

[112] Klementiev, A., Titov, I., & Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012* (pp. 1459–1474).

[113] Knight, K. & Graehl, J. (1998). Machine transliteration. *Computational linguistics*, 24(4), 599–612.

[114] Knight, K., Nair, A., Rathod, N., & Yamada, K. (2006). Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions* (pp. 499–506).

[115] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions* (pp. 177–180).: Association for Computational Linguistics.

[116] Koehn, P., Och, F. J., & Marcu, D. (2003). 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* (pp. 48–54).: Association for Computational Linguistics.

[117] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

[118] Kulkarni, V., Al-Rfou, R., Perozzi, B., & Skiena, S. (2015). Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 625–635).

[119] Kumari, A. & Goyal, V. (2012). Font convertors for indian languages-a survey. *Computer Science*, 1(12).

[120] Kundu, S., Paul, S., & Pal, S. (2018). A deep learning based approach to transliteration. In *Proceedings of the Seventh Named Entities Workshop* (pp. 79–83).

[121] Kuo, J.-S., Li, H., & Yang, Y.-K. (2006). Learning transliteration lexicons from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 1129–1136).: Association for Computational Linguistics.

[122] Kvapilıková, I., Artetxe, M., Labaka, G., Agirre, E., & Bojar, O. (2021). Unsupervised multilingual sentence embeddings for parallel corpus mining. *arXiv preprint arXiv:2105.10419*.

[123] Laitonjam, L., Singh, L. G., & Singh, S. R. (2018). Transliteration of english loanwords and named-entities to manipuri: Phoneme vs grapheme representation. In *2018 International Conference on Asian Language Processing (IALP)* (pp. 255–260).: IEEE.

[124] Laitonjam, L. & Singh, S. R. (2021). Manipuri-english machine translation using comparable corpus. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)* (pp. 78–88).

[125] Lakew, S. M., Lotito, Q. F., Negri, M., Turchi, M., & Federico, M. (2017). Improving zero-shot translation of low-resource languages. In *14th International Workshop on Spoken Language Translation*.

[126] Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. (2018a). Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

[127] Lample, G., Ott, M., Conneau, A., Denoyer, L., & Ranzato, M. (2018b). Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

[128] Le, N. T. & Sadat, F. (2018). Low-resource machine transliteration using recurrent neural networks of asian languages. In *Proceedings of the Seventh Named Entities Workshop* (pp. 95–100).

[129] Le, N. T., Sadat, F., Menard, L., & Dinh, D. (2019). Low-resource machine transliteration using recurrent neural networks. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(2), 13.

[130] Lehal, G. S., Saini, T. S., & Buttar, S. P. K. (2014). Automatic bilingual legacy-fonts identification and conversion system. *Res. Comput. Sci.*, 86, 9–23.

[131] Leng, Y., Tan, X., Qin, T., Li, X.-Y., & Liu, T.-Y. (2019). Unsupervised pivot translation for distant languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 175–183).

[132] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10 (pp. 707–710).

[133] Levy, O., Søgaard, A., & Goldberg, Y. (2016). A strong baseline for learning cross-lingual word embeddings from sentence alignments. *arXiv preprint arXiv:1608.05426*.

[134] Libovickỳ, J. & Helcl, J. (2017). Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 196–202).

[135] Libovickỳ, J., Helcl, J., & Mareček, D. (2018). Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers* (pp. 253–260).

[136] Lu, A., Wang, W., Bansal, M., Gimpel, K., & Livescu, K. (2015). Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 250–256).

[137] Lü, L., Zhang, Z.-K., & Zhou, T. (2013). Deviation of zipf's and heaps' laws in human languages with limited dictionary sizes. *Scientific reports*, 3(1), 1–7.

[138] Luong, M.-T., Kayser, M., & Manning, C. D. (2015a). Deep neural language models for machine translation. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning* (pp. 305–309).

[139] Luong, M.-T., Pham, H., & Manning, C. D. (2015b). Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing* (pp. 151–159).

[140] Luong, M.-T., Pham, H., & Manning, C. D. (2015c). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[141] Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2014). Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

[142] Maimaiti, M., Liu, Y., Luan, H., & Sun, M. (2019). Multi-round transfer learning for low-resource nmt using multiple high-resource languages. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(4), 1–26.

[143] Marchisio, K., Duh, K., & Koehn, P. (2020). When does unsupervised machine translation work? In *Proceedings of the Fifth Conference on Machine Translation* (pp. 571–583).

[144] Marie, B. & Fujita, A. (2020). Iterative training of unsupervised neural and statistical machine translation systems. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(5), 1–21.

[145] Meitei, S. P., Purkayastha, B. S., & Devi, H. M. (2015). Development of a manipuri stemmer: A hybrid approach. In *Advanced Computing and Communication (ISACC), 2015 International Symposium on* (pp. 128–131).: IEEE.

[146] Meng, H. M., Lo, W.-K., Chen, B., & Tang, K. (2001). Generating phonetic cognates to handle named entities in english-chinese cross-language spoken document retrieval. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.* (pp. 311–314).: IEEE.

[147] Merhav, Y. & Ash, S. (2018). Design challenges in named entity transliteration. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 630–640).

[148] Mikolov, T., Le, Q. V., & Sutskever, I. (2013a). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

[149] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2* (pp. 3111–3119).

[150] Mrkšić, N., Vulić, I., Séaghdha, D. Ó., Leviant, I., Reichart, R., Gašić, M., Korhonen, A., & Young, S. (2017). Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the association for Computational Linguistics*, 5, 309–324.

[151] Munteanu, D. S. & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4), 477–504.

[152] Naim, I., Riley, P., & Gildea, D. (2018). Feature-based decipherment for machine translation. *Computational Linguistics*, 44(3), 525–546.

[153] Ngo, G. H., Nguyen, M., & Chen, N. F. (2019). Phonology-augmented statistical framework for machine transliteration using limited linguistic resources. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 27(1), 199–211.

[154] Nguyen, D., Overwijk, A., Hauff, C., Trieschnigg, D. R., Hiemstra, D., & De Jong, F. (2008). Wikitranslate: query translation for cross-lingual information retrieval using only wikipedia. In *Workshop of the Cross-Language Evaluation Forum for European Languages* (pp. 58–65).: Springer.

[155] Nguyen, T. Q. & Chiang, D. (2017). Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 296–301).

[156] Nicolai, G., Hauer, B., Salameh, M., St Arnaud, A., Xu, Y., Yao, L., & Kondrak, G. (2015). Multiple system combination for transliteration. In *Proceedings of the Fifth Named Entity Workshop* (pp. 72–77).

[157] Nirenburg, S. (1989). Knowledge-based machine translation. *Machine Translation*, 4(1), 5–24.

[158] Nirmal, Y. & Sharma, U. (2018). Problems and issues in parsing manipuri text. In *Proceedings of the International Conference on Computing and Communication Systems* (pp. 393–401).: Springer.

[159] Nirmal, Y. & Sharma, U. (2019). A grammar-driven approach for parsing manipuri language. In *International Conference on Pattern Recognition and Machine Intelligence* (pp. 267–274).: Springer.

[160] Nishimura, Y., Sudoh, K., Neubig, G., & Nakamura, S. (2018). Multi-source neural machine translation with missing data. *arXiv preprint arXiv:1806.02525*.

[161] Nongmeikapam, K., Salam, B., Romina, M., Chanu, N. M., & Bandyopadhyay, S. (2011a). A light weight manipuri stemmer. In *Proc. National Conference on Indian Language, Computing (NCILC)*.

[162] Nongmeikapam, K., Singh, N. H., Thoudam, S., & Bandyopadhyay, S. (2011b). Manipuri transliteration from bengali script to meitei mayek: A rule based approach. In *Information Systems for Indian Languages* (pp. 195–198). Springer.

[163] Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics* (pp. 160–167).

[164] Oh, J., Choi, K., & Isahara, H. (2006a). A comparison of different machine transliteration models. *Journal of Artificial Intelligence Research*, 27, 119–151.

[165] Oh, J.-H., Choi, K.-S., & Isahara, H. (2006b). A machine transliteration model based on correspondence between graphemes and phonemes. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(3), 185–208.

[166] Paik, J. H., Mitra, M., Parui, S. K., & Järvelin, K. (2011). Gras: An effective and efficient stemming algorithm for information retrieval. *ACM Transactions on Information Systems (TOIS)*, 29(4), 19.

[167] Pan, S. J. & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345–1359.

[168] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318).: Association for Computational Linguistics.

[169] Pham, N.-Q., Niehues, J., Ha, T.-L., & Waibel, A. (2019). Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)* (pp. 13–23).

[170] Popović, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation* (pp. 392–395).

[171] Popović, M. (2017). chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation* (pp. 612–618).

[172] Quirk, C., Menezes, A., & Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 271–279).: Association for Computational Linguistics.

[173] Raganato, A., Tiedemann, J., et al. (2018). An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*: Association for Computational Linguistics.

[174] Rahaman, S. S., Islam, M. R., Akhand, M., et al. (2013). Design and development of a bengali unicode font converter. In *2013 International Conference on Informatics, Electronics and Vision (ICIEV)* (pp. 1–4).: IEEE.

[175] Rahul, L., Nandakishor, S., Singh, L. J., & Dutta, S. (2013). Design of manipuri keywords spotting system using hmm. In *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)* (pp. 1–3).: IEEE.

[176] Raj, A. A. & Maganti, H. (2009). Transliteration based search engine for multilingual information access. In *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3)* (pp. 12–20).

[177] Raj, A. A. & Prahallad, K. (2007). Identification and conversion of font-data in indian languages. In *In International Conference on Universal Digital Library (ICUDL)*.

[178] Rama, T. & Gali, K. (2009). Modeling machine transliteration as a phrase based statistical machine translation problem. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)* (pp. 124–127).

[179] Rapp, R., Sharoff, S., & Zweigenbaum, P. (2016). Recent advances in machine translation using comparable corpora. *Natural Language Engineering*, 22(4), 501–516.

[180] Rapp, R., Zweigenbaum, P., & Sharoff, S. (2020). Overview of the fourth bucc shared task: Bilingual dictionary induction from comparable corpora. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora* (pp. 6–13).

[181] Ravi, S. & Knight, K. (2011). Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 12–21).

[182] Ren, S., Wu, Y., Liu, S., Zhou, M., & Ma, S. (2019a). Explicit cross-lingual pre-training for unsupervised machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 770–779).

[183] Ren, S., Zhang, Z., Liu, S., Zhou, M., & Ma, S. (2019b). Unsupervised neural machine translation with smt as posterior regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33 (pp. 241–248).

[184] Richardson, L. (2007). Beautiful soup documentation. *Dosegljivo: https://www. crummy. com/software/BeautifulSoup/bs4/doc/.[Dostopano: 7. 7. 2018]*.

[185] Rosca, M. & Breuel, T. (2016). Sequence-to-sequence neural network models for transliteration. *arXiv preprint arXiv:1610.09565*.

[186] Ruder, S., Vulić, I., & Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65, 569–631.

[187] Rudra Murthy, V., Kunchukuttan, A., & Bhattacharyya, P. (2019). Addressing word-order divergence in multilingual neural machine translation for extremely low resource languages. In *Proceedings of NAACL-HLT* (pp. 3868–3873).

[188] Sabbah, F. & Aker, A. (2018). Creating comparable corpora through topic mappings. In *the 11th Workshop on BUCC at LREC 2018* (pp. 19–24).

[189] Saikia, R. & Singh, S. R. (2016). Generating manipuri english pronunciation dictionary using sequence labelling problem. In *Asian Language Processing (IALP), 2016 International Conference on* (pp. 67–70).: IEEE.

[190] Sato, S. & Nagao, M. (1990). Toward memory-based translation. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*, COLING '90 (pp. 247–252). Stroudsburg, PA, USA: Association for Computational Linguistics.

[191] Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1), 1–10.

[192] Schroeder, J., Cohn, T., & Koehn, P. (2009). Word lattices for multi-source translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)* (pp. 719–727).

[193] Schuster, M. & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681.

[194] Schwenk, H. (2018). Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 228–234).

[195] Schwenk, H., Chaudhary, V., Sun, S., Gong, H., & Guzmán, F. (2021). Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 1351–1361).

[196] Sellami, R., Sadat, F., & Beluith, L. H. (2018). Building and exploiting domain-specific comparable corpora for statistical machine translation. In *Intelligent Natural Language Processing: Trends and Applications* (pp. 659–676).: Springer.

[197] Sennrich, R., Haddow, B., & Birch, A. (2015a). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

[198] Sennrich, R., Haddow, B., & Birch, A. (2015b). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

[199] Sennrich, R., Haddow, B., & Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 86–96).

[200] Sharma, H. S. (1999). A comparison between khasi and manipuri word order. *Linguistics of the Tibeto-Burman Area*, 22(1), 139–48.

[201] Sharoff, S., Rapp, R., & Zweigenbaum, P. (2013). Overviewing important aspects of the last twenty years of research in comparable corpora. In *Building and Using Comparable Corpora* (pp. 1–17).: Springer.

[202] Shchukin, V., Khristich, D., & Galinskaya, I. (2016). Word clustering approach to bilingual document alignment (wmt 2016 shared task). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers* (pp. 740–744).

[203] Sing, T. D. & Bandyopadhyay, S. (2010). Statistical machine translation of english–manipuri using morpho-syntactic and semantic information. *Proceedings of the Association for Machine Translation in the Americas (AMTA 2010)*.

[204] Singh, L. G., Laitonjam, L., & Singh, S. R. (2016). Automatic syllabification for manipuri language. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 349–357).

[205] Singh, L. S., Thaoroijam, K., & Das, P. K. (2007). Written manipuri (meiteiron) from phoneme to grapheme. *Language in India*, 7(6).

[206] Singh, R. & Singh, S. (2021). Text similarity measures in news articles by vector space model using nlp. *Journal of The Institution of Engineers (India): Series B*, 102(2), 329–338.

[207] Singh, S. M. & Singh, T. D. (2020). Unsupervised neural machine translation for english and manipuri. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages* (pp. 69–78).

[208] Singh, T. D. (2012a). Bidirectional bengali script and meetei mayek transliteration of web based manipuri news corpus. In *the Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP) of COLING* (pp. 181–189).

[209] Singh, T. D. (2012b). Building parallel corpora for smt system: A case study of english-manipuri. *International Journal of Computer Applications*, 52(14).

[210] Singh, T. D. (2013). Taste of two different flavours: Which manipuri script works better for english-manipuri language pair smt systems? In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation* (pp. 11–18).

[211] Singh, T. D. & Bandyopadhyay, S. (2005). Manipuri morphological analyzer. In *In the Proceedings of the Platinum Jubilee International Conference of LSI, Hyderabad, India*.

[212] Singh, T. D. & Bandyopadhyay, S. (2006). Word class and sentence type identification in manipuri morphological analyzer,". *In Proceedings of MSPIL, Mumbai, India*, (pp. 11–17).

[213] Singh, T. D. & Bandyopadhyay, S. (2008). Morphology driven manipuri pos tagger. In *Proceedings of the IJCNLP-08 Workshop on NLP for less privileged languages*.

[214] Singh, T. D. & Bandyopadhyay, S. (2010). Manipuri-english example based machine translation system. *International Journal of Computational Linguistics and Applications (IJCLA), ISSN*, (pp. 0976–0962).

[215] Singh, T. D. & Bandyopadhyay, S. (2011). Integration of reduplicated multiword expressions and named entities in a phrase based statistical machine translation system. In *Proceedings of 5th international joint conference on natural language processing* (pp. 1304–1312).

[216] Singh, T. D., Ekbal, A., & Bandyopadhyay, S. (2008). Manipuri pos tagging using crf and svm: A language independent approach. In *proceeding of 6th International conference on Natural Language Processing (ICON-2008)* (pp. 240–245).

[217] Singh, T. D., Nongmeikapam, K., Ekbal, A., & Bandyopadhyay, S. (2009). Named entity recognition for manipuri using support vector machine. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2* (pp. 811–818).

[218] Singha, K. R., Purkayastha, B. S., & Singha, K. D. (2012). Part of speech tagging in manipuri: A rule based approach. *International Journal of Computer Applications*, 51(14).

[219] Singhania, S., Nguyen, M., Ngo, G. H., & Chen, N. (2018). Statistical machine transliteration baselines for news 2018. In *Proceedings of the Seventh Named Entities Workshop* (pp. 74–78).

[220] Siripragada, S., Philip, J., Namboodiri, V. P., & Jawahar, C. (2020). A multilingual parallel corpora collection effort for indian languages. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 3743–3751).

[221] Skadiņa, I., Gaizauskas, R., Babych, B., Ljubešić, N., Tufiş, D., & Vasiļjevs, A. (2019). *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*. Springer.

[222] Smith, J., Quirk, C., & Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 403–411).

[223] Søgaard, A., Ruder, S., & Vulić, I. (2018). On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 778–788).

[224] Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2019). Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

[225] Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

[226] Štajner, T. & Mladenić, D. (2019). Cross-lingual document similarity estimation and dictionary generation with comparable corpora. *Knowledge and Information Systems*, 58(3), 729–743.

[227] Sun, H., Wang, R., Utiyama, M., Marie, B., Chen, K., Sumita, E., & Zhao, T. (2021a). Unsupervised neural machine translation for similar and distant language pairs: An empirical study. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 20(1), 1–17.

[228] Sun, Y., Zhu, S., Yifan, F., & Mi, C. (2021b). Parallel sentences mining with transfer learning in an unsupervised setting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop* (pp. 136–142).

[229] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

[230] Tang, G., Cap, F., Pettersson, E., & Nivre, J. (2018a). An evaluation of neural machine translation models on historical spelling normalization. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1320–1331).

[231] Tang, G., Müller, M., Gonzales, A. R., & Sennrich, R. (2018b). Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4263–4272).

[232] Tennage, P., Sandaruwan, P., Thilakarathne, M., Herath, A., Ranathunga, S., Jayasena, S., & Dias, G. (2017). Neural machine translation for sinhala and tamil languages. In *2017 International Conference on Asian Language Processing (IALP)* (pp. 189–192).: IEEE.

[233] Trieu, H.-L., Tran, D.-V., Ittoo, A., & Nguyen, L.-M. (2019). Leveraging additional resources for improving statistical machine translation on asian low-resource languages. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3), 1–22.

[234] Tsou, B. K. & Chow, K. (2019). From the cultivation of comparable corpora to harvesting from them: A quantitative and qualitative exploration. In *Proceedings of the Conference on Building and Using Comparable Corpora (BUCC 2019)* (pp. 29–36).

[235] Udupa, R., Saravanan, K., Kumaran, A., & Jagarlamudi, J. (2009). Mint: A method for effective and scalable mining of named entity transliterations from large comparable corpora. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 799–807).: Association for Computational Linguistics.

[236] Utiyama, M. & Isahara, H. (2007). A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Associa-*

tion for Computational Linguistics; Proceedings of the Main Conference (pp. 484–491).

[237] van Leijenhorst, D. C. & Van der Weide, T. P. (2005). A formal derivation of heaps' law. *Information Sciences*, 170(2-4), 263–272.

[238] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

[239] Vilar, D., Stein, D., & Ney, H. (2008). Analysing soft syntax features and heuristics for hierarchical phrase based machine translation. In *Proceedings of the 5th International Workshop on Spoken Language Translation: Papers*.

[240] Virga, P. & Khudanpur, S. (2003). Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition-Volume 15* (pp. 57–64).: Association for Computational Linguistics.

[241] Vulić, I. & Korhonen, A. (2016). On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 247–257).

[242] Vulić, I. & Moens, M.-F. (2015). Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 363–372).

[243] Wloka, B. (2018). Identifying bilingual topics in wikipedia for efficient parallel corpus extraction and building domain-specific glossaries for the japanese-english language pair. In *11th Workshop on Building and Using Comparable Corpora* (pp.15).

[244] Wu, H. & Wang, H. (2007). Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3), 165–181.

[245] Wu, L., Zhu, J., He, D., Gao, F., Qin, T., Lai, J., & Liu, T.-Y. (2019). Machine translation with weakly paired documents. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 4375–4384).

[246] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[247] Yasuda, K. & Sumita, E. (2008). Method for building sentence-aligned corpus from wikipedia. In *2008 AAAI Workshop on Wikipedia and Artificial Intelligence (WikiAI08)* (pp. 263–268).

[248] Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8(1), 59–60.

[249] Zhang, Y., Jatowt, A., Bhowmick, S. S., & Tanaka, K. (2016). The past is not a foreign country: Detecting semantically similar terms across time. *IEEE Transactions on Knowledge and Data Engineering*, 28(10), 2793–2807.

[250] Zhu, S., Yang, Y., & Xu, C. (2020). Extracting parallel sentences from nonparallel corpora using parallel hierarchical attention network. *Computational Intelligence and Neuroscience*, 2020.

[251] Zipf, G. K. (1949). Human behaviour and the principle of least-effort. *Cambridge MA edn, Reading: Addison-Wesley*.

[252] Zoph, B. & Knight, K. (2016). Multi-source neural translation. In *Proceedings of NAACL-HLT* (pp. 30–34).

[253] Zoph, B., Yuret, D., May, J., & Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1568–1575).

[254] Zweigenbaum, P., Sharoff, S., & Rapp, R. (2016). Towards preparation of the second bucc shared task: Detecting parallel sentences in comparable corpora. In *Proceedings of the Ninth Workshop on Building and Using Comparable Corpora. European Language Resources Association (ELRA), Portoroz, Slovenia* (pp. 38–43).

Publications

- JOURNALS

  1. **Lenin Laitonjam**, Sanasam Ranbir Singh. **A Hybrid Machine Transliteration Model Based on Multi-source Encoder–Decoder Framework: English to Manipuri**. *SN Computer Science*. 3, 125 (2022). https://doi.org/10.1007/s42979-021-01005-9
  2. **Lenin Laitonjam** and Sanasam Ranbir Singh. **Manipuri-English Comparable Corpus for Cross-lingual studies**. *Language Resources and Evaluation*. 1-37 (2022). https://doi.org/10.1007/s10579-021-09576-y
  3. **Lenin Laitonjam** and Sanasam Ranbir Singh. **A Machine Translation System using Temporal-Aligned Comparable Corpus: A case study for Manipuri-English**. [Under Review]
  4. **Lenin Laitonjam** and Sanasam Ranbir Singh. **Effect of document-aligned Comparable Corpus on Manipuri-English Machine Translation**. [Under Review]

- CONFERENCES

  1. **Lenin Laitonjam**, Loitongbam Gyanendro Singh, and Sanasam Ranbir Singh. **Transliteration of English Loanwords and Named-entities to Manipuri: Phoneme vs Grapheme representation**. In *Proceedings of the 2018 International Conference on Asian Language Processing (IALP)*, IALP 2018, Bandung, Indonesia, pages 255-260. doi:10.1109/IALP.2018.8629141.
  2. **Laitonjam, Lenin**, and Sanasam Ranbir Singh. **Manipuri-English Machine Translation using Comparable Corpus**. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*, pages. 78-88.
  3. **Lenin Laitonjam**, and Sanasam Ranbir Singh. **Manipuri-English Cross-lingual Word Embeddings using a Temporally Aligned Comparable Corpus**. In *Proceedings of the 2021 International Conference on Asian*

Language Processing (IALP), Singapore, IALP 2021, pages 195-199, doi:10.1109/IALP54817.2021.9675204.

8.2.1 Other Publications (Not Related to Thesis Work)

1. Loitongbam Gyanendro Singh, Lenin Laitonjam, and Sanasam Ranbir Singh. **Automatic Syllabification for Manipuri Language**. In *Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics*, Association for Computational Linguistics 2016 pages 349-357.