APPROACHES TO HANDLE SCARCE RESOURCES FOR BENGALI STATISTICAL MACHINE TRANSLATION

by

Maxim Roy
B.Sc., University of Windsor, 2002
M.Sc., University of Windsor, 2005

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Spring 2010

APPROVAL

Name: Maxim Roy

Degree: Doctor of Philosophy

Title of Thesis: Approaches to handle scarce resources for Bengali Statistical Machine Translation

Examining Committee: Dr. Greg Mori, Assistant Professor, SFU
Chair

Dr. Fred Popowich, Senior Supervisor
Professor, Computing Science
Simon Fraser University

Dr. Anoop Sarkar, Supervisor
Associate Professor, Computing Science
Simon Fraser University

Dr. Paul McFetridge, Internal Examiner
Associate Professor, Linguistics
Simon Fraser University

Dr. Greg Kondrak, External Examiner
Associate Professor, Computer Science
University of Alberta

Date Approved: April 15, 2010

Abstract

Machine translation (MT) is a hard problem because of the highly complex, irregular and diverse nature of natural language. MT refers to computerized systems that utilize software to translate text from one natural language into another with or without human assistance. It is impossible to accurately model all the linguistic rules and relationships that shape the translation process, and therefore MT has to make decisions based on incomplete data. In order to handle this incomplete data, a principled approach is to use statistical methods to make optimum decisions given incomplete data. Statistical machine translation (SMT) uses a probabilistic framework to automatically translate text from one language to another. Using the co-occurrence counts of words and phrases from the bilingual parallel corpora where sentences are aligned with their translation, SMT learns the translation of words and phrases.

We apply SMT techniques for translation between Bengali and English. SMT systems requires a significant amount of bilingual data between language pairs to achieve significant translation accuracy. However, being a low-density language, such resources are not available in Bengali. So in this thesis, we investigate different language independent and dependent techniques which can be helpful to improve translate accuracy of Bengali SMT systems.

We explore the transliteration module, prepositional and Bengali compound word handling module in the context of Bengali to English SMT. Further we look into semi-supervised techniques and active learning techniques in Bengali SMT to deal with scarce resources.

Also due to different word orders in Bengali and English, we also propose different syntactic phrase reordering techniques for Bengali SMT. We also contributed toward Bengali SMT by creating a new test set, lexicon and by developing Bengali text processing tools such as tokenizer, sentence segmenter, and morphological analyzer.

Overall the main objective of this thesis is to make a contribution towards Bengali

language processing, provide a general foundation for conducting research in Bengali SMT and improve the quality of Bengali SMT.

I dedicate this thesis to my lovely wife, Sicily

"If you knew what you were doing it wouldn't be called research"

— *Einstein*

Acknowledgments

First and foremost I want to thank my advisor Prof. Fred Popowich. I would like to express my deep gratitude for his encouragement, guidance and support from the initial to the final stage of my Ph.D. degree. He had a profound influence on my research and studies and it has been an honor to be his Ph.D. student. I also want to express my deeply-felt thanks to my other advisor Prof. Anoop Sarkar who introduced me to the fascinating field of Statistical Machine Translation, and taught me a great deal of valuable research skills. I would like to thank the members of my thesis committee: Prof. Paul McFetridge, Prof. Greg Kondrak, and Prof. Greg Mori for their valuable comments and feedbacks on my thesis.

I am thankful to the past and current members of Natural Language Lab at the School of Computing Sciences at Simon Fraser University: Ann Clifton, Chris Demwell, Akshay Gattani, Ajeet Grewal, Baohua Gu, Gholamreza Haffari, Fahim Hasan, Yudong Liu, Stephen Makonin, Mehdi Mostafavi-Kashani, Majid Razmara, Baskaran Sankaran, Asqar Saypil, Dong Song, Milan Tofiloski, Zhongmin Shi, Yang (Wendy) Wang, and Winona Wu. During my graduate studies, I have had the great opportunity to collaborate with my labmate Gholamreza Haffari from whom I also learnt valuable research skills.

Finally, I would like to thank my family: my parents Manindra Kumar Roy and Olga Roy, for giving me life in the first place, for raising me with a love for science and for their unconditional support and encouragement in all my pursuits. My special gratitude to my brother Roman, my in-laws and my sister-in-law Sanita for their loving support. Last but not the least, I am indebted to my lovely wife, Sicily. I would like to thank her for the many sacrifices she has made to support me in undertaking my doctoral studies. Her support and encouragement has seen me through tumultuous times. Without her support, it would have been impossible for me to finish this work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years due to increasing cross-regional communication and the need for information exchange, the demand for language translation has greatly increased. It is becoming hard for professional translators to meet the increasing demands of translation. The assistance of computers can be used as a substitute in such a situation [53]. However, computers do not have the ability to deal adequately with the various complexities of language that humans handle naturally such as ambiguity, syntactic irregularity, multiple word meanings and the influence of context. A classic example[1] is illustrated in the following pair of sentences:

$$\text{Time flies like an arrow.} \tag{1.1}$$

$$\text{Fruit flies like a banana.} \tag{1.2}$$

The sentence construction is parallel, but the meanings are entirely different. The first sentence (1.1) is a figure of speech involving a simile and the second sentence (1.2) is a literal description, and the identical words in the sentences - “flies” and “like” - are used in different grammatical categories. A computer can be programmed to understand either of these examples, but it is difficult to distinguish between them. It is also very hard for computers to translate between languages because of these various complexities of language.

Machine translation (MT) refers to computerized systems that utilize computer software to translate text from one natural language into another with or without human assistance.

[1] http://www.diplomacy.edu/language/Translation/machine.htm

Human-level translation quality might be ideal but is hard to achieve. The complexity of MT is due to the factors such as 1) the cultural barriers between different languages, 2) the inherent ambiguity of human languages and 3) the irregularity between languages.

The cultural barriers between different languages: Cultures provide people with ways of thinking, seeing, hearing, and interpreting the world. Thus the same words can mean different things to people from different cultures, even when they speak the "same" language. For English speakers from two different regions, there might be moments of misunderstanding because of cultural differences that assign different words to different things or assign different meaning to similar phrases. Also for example in Bengali, people of different regions might use different words to express the same word "grandma". When the languages are different, and translation has to be used to communicate, the potential for misunderstandings increases and the MT task becomes more difficult.

The inherent ambiguity of human languages: Most natural language words are subject to ambiguity and polysemy. A word, phrase or a sentence that has more than one meaning is said to be ambiguous. There are several types of ambiguity. Lexical ambiguity is associated with polysemy and homonymy. Words that are identical in sound and spelling are called full homonyms, e.g. ball (a round object used in games) and ball (a gathering of people for dancing). A case of polysemy is one where a word has several clearly related senses, e.g. mouth (of a river vs. of an animal). Syntactic ambiguity is the result of different grouping of elements, and it results from the different logical relationships between elements. These inherent ambiguities of most natural languages make the MT task hard.

Irregularity between languages: Many languages are characterized by certain morphological irregularities. Most European languages are famous for their irregular verbs. English is certainly no exception. For examples in English there is "boot" and "boots" but not "foot" and "foots" nor "boot" and "beet". Other examples are "link" and "linked" but not "think" and "thinked" as in "sing, sang, sung" nor "link, lank, lunk".

In MT, from the beginning, researchers have focused on technical and news documents, which simplify the problem a little since these documents contain fewer cultural differences. A major difficulty in MT of one natural language to another is due to the varied structures and lexical choices for the same concept in different languages.

What makes the MT task an interesting and challenging problem? Natural languages are highly complex. A lot of words have different meanings and also different translations. Similar sentences might have different meanings in different contexts and the relationship between linguistic entities might sometimes be vague. Some grammatical relations in one language might not apply to another language and sentences involving these relations need to be significantly reformulated. For certain languages such as Chinese or Japanese, even word boundaries don't exist. So to develop an MT system, many such dependencies need to be taken into account. Sometimes such dependencies are weak and vague, which makes it even hard to describe simple and relevant rules in the translation process. We have to consider various such dependencies such as morphological, syntactic, semantic and pragmatic dependencies.

Overall, MT is a hard problem because of the highly complex, irregular and diverse nature of natural language. It is impossible to accurately model all the linguistic rules and relationships that shape the translation process, and therefore MT has to make decisions based on incomplete data. In order to handle this incomplete data, a principled approach to this problem is to use statistical methods to make an appropriate decisions given incomplete data.

1.1 Statistical Machine Translation

The idea of using statistical techniques in MT is not new. In 1955, Warren Weaver suggested using statistical and cryptographic techniques from the then emerging field of communication theory to the problem of translating text from one language to another. Due to various reasons the research in this direction was soon abandoned [52]. Some important factors were the lack of computational power and the lack of machine-readable parallel texts from which the statistics are usually gathered. Instead, the focus turned to rule based and knowledge-based systems, some of which have seen commercial success for major language pairs like English and French. One of the limitations of rule based and knowledge-based systems is that it can take a considerable amount of time to develop the resources required to translate between just a single language pair using translation rules and lexicons for MT. Also, it does not have a probabilistic framework to handle unforeseen situations.

In the early 1990's, statistical machine translation(SMT) systems emerged due to the availability of bilingual corpora and also increasingly powerful computers. SMT uses a probabilistic framework to automatically translate text from one language to another. Using the co-occurrence counts of words and phrases from the bilingual parallel corpora where sentences are aligned with their translation, SMT learns the translation of words and phrases.

As in other fields of artificial intelligence, it is impossible to accurately model the highly complex processes involved. SMT provides a rigorous mathematical framework within which decisions can be made about how best to translate a sentence.

1.2 Challenges in Bengali-English Machine Translation

SMT techniques offer great promise for the development of automatic translation systems. However, the realization of this potential requires the availability of significant amount of bilingual corpora between language pairs. For some resource scarce languages such as Bengali(see details in chapter 2) these resources are not available. The acquisition of large high-quality bilingual parallel corpora between such language pairs requires significant time and effort. We are therefore studying methods to better exploit existing training data (bilingual corpora) with the prospect of building SMT systems between Bengali and English.

MT between Bengali and English possesses its own challenges. We are translating to English from a morphologically rich language, Bengali. Bengali grammar generally follows the Subject Object Verb (SOV) structure and English follows the Subject Verb Object (SVO) structure. Bengali is not only a morphologically rich language but also considered as a ”low-density” or resource scare language. Languages are called “low-density”, either because the population speaking the language is not very large, or even if millions of people speak the language, insufficient amounts of parallel text are available in that language. Since SMT systems generally heavily rely on a large amount of parallel corpora we are also focusing on the development of techniques which improve translation accuracy with limited amount of available resources. Also research work in the field of Bengali SMT is still in an early stage.

Our main research focus is in the development of methods and techniques for SMT for resource scarce languages such as Bengali. In this thesis while exploring different techniques in SMT we will focus on Bengali as the source language and English as the target language. We make several novel contributions with respect to Bengali SMT.

Word reordering approach: We developed several word-reordering techniques, which reorder the source language, to make it closer to target language structure before applying the translation process.

Semi-supervised approach: We applied several semi-supervised techniques [107], which effectively used monolingual corpora of the source language, which is Bengali, together with bilingual corpora between Bengali and English to improve translation accuracy.

Active learning approach: We investigated active learning [46] in Bengali-English SMT and experimented with several sentence selection strategies to select sentences from monolingual Bengali corpora to improve quality.

Bengali specific components: In order to handle transliteration, prepositional and compound words in Bengali we developed three Bengali specific components which can be added to Bengali SMT system to improve translation accuracy.

Bengali tools: As part of the preprocessing step for Bengali SMT we developed several Bengali text processing tools including a tokenizer, sentence segmenter, and morphological analyzer. These tools were designed for SMT but are useful for other NLP applications.

New test set: We developed test sets for evaluation of the Bengali compound word splitting task and the transliteration task.

Bengali lexicon: We developed a Bengali lexicon by collecting different online resources which is part of different components of our SMT system.

Extended SMT evaluation test set: As part of the SMT evaluation we have also extended the Linguistic Data Consortium(LDC) test set by adding two additional reference translation sets between Bengali and English.

New Manual evaluation: We proposed a new manual evaluation approach for the MT community and evaluate our components using the new manual evaluation approach.

1.3 Thesis Overview

This thesis describes methods and techniques for SMT from a resource scarce language, Bengali to English. In chapter 2, we provide a brief introduction to the background of SMT. In chapter 3, we present an overview of the Bengali language and the Bengali dataset. In chapter 4, we introduce our Bengali specific transliteration, compound splitting, prepositional module and word reordering techniques. In chapter 5, we describe semi-supervised and active learning techniques for Bengali SMT. And finally in chapter 6 we provide concluding comments and some future directions of this research work.

Chapter 2

Machine Translation

Machine translation (MT) is probably the oldest application of natural language processing. Its more than 55 years of history have seen the development of several major approaches. Despite the commercial success of many MT systems, tools, and other products, the general problem remains unsolved, and the various ways of combining approaches and paradigms are only beginning to be explored.

MT has been defined as the process of translating text units of one language (source language) into a second language (target language) by using computers. MT is a hard problem because of the highly complex, irregular and diverse nature of natural languages. A good historical overview of MT can be found in [52], and a comprehensive general survey in [79].

Statistical machine translation (SMT) refers to a subset of MT approaches that are characterized by their use of machine learning methods. Although the first systems were developed less than two decades ago, SMT currently dominates the research field. From the initial word-based translation models [13], research on SMT has seen dramatic improvement. At the end of the last decade the use of context in the translation model which is known as a phrase-based MT approach led to a clear improvement in translation quality ([138], [135], [82]).

2.1 Statistical Machine Translation

The use of statistical data for MT has been suggested since the age of first generation MT. However, this approach was not pursued extensively. This is mainly due to the fact that computers in those days were not powerful enough to support such a computationally intensive approach.

The idea behind the pure statistical MT approach is to let a computer learn automatically how to translate text from one language to another by examining large amounts of parallel bilingual text, i.e. documents which are nearly exact translations of each other. The statistical MT approach uses statistical data to perform translation. This statistical data is obtained from an analysis of a vast amount of bilingual texts. SMT applies a learning algorithm to a large body of previously translated text, known variously as a parallel corpus, parallel text, bitext, or multitext. The learner is then able to translate previously unseen sentences. With an SMT toolkit and enough parallel text, we can build an MT system for a new language pair within a very short period of time perhaps as little as a day ([4], [94]). The goal of SMT is to translate an input word sequence from the source language into a target language word sequence. Given the source language sequence, we should choose the target language sequence which maximizes the posterior probability.

SMT converts a source language text (f) into a target language text (e) according to the following formula:

$$e = argmax_e p(e|f) \tag{2.1}$$

Bayes Rule can be applied to the above to derive:

$$e = argmax_e p(f|e)p(e) \tag{2.2}$$

The translation process is treated as a noisy channel model, like those used in speech recognition in which there exists e transcribed as f, and a translation is to determine the best e from f in terms of $p(f|e)p(e)$. The term, $p(f|e)$, is a translation model representing some correspondence between bilingual text and $p(e)$ is the language model. In addition, a word correspondence model, called alignment a, is introduced to the translation model to represent a positional correspondence of the target and source words:

$$e = argmax_e p(f,a|e)p(e) \tag{2.3}$$

Most of the earlier SMT systems were single-word based approaches where each word in the source language is aligned to exactly one word in the target language in the translation model. One major disadvantage of single-word based approaches is that the contextual information is not taken into account. The lexicon probabilities are based only on single words. For many words, the translation depends heavily on the surrounding words which a single-word based translation approach is not capable of handling.

2.1.1 Word-Based Models

The first machine translation systems based on probabilistic translation models [13] are known as word-based models. These models were created from the original work by the IBM Candide project on SMT in the late 1980s and early 1990s. A simple word-based model for machine translation is solely based on lexical translation, which translates words in isolation. Word-based models also known as IBM models follow an approach to machine translation that is based on statistics collected over a parallel corpus of translated text. The models break up the translation process into a number of small steps for which sufficient statistics can be collected. This method is known as generative modelling.

The IBM models are based on word translation that is modelled by a lexical translation probability distribution. Maximum Likelihood Estimation(MLE) is used to train this distribution from data by collecting counts for word translations. The information about alignment between the input and the output words, which is missing in the data, is a hidden variable in the model. The Expectation Maximization(EM) algorithm handles the problem of incomplete data and perplexity is used to measure how well the model fits the data.

The IBM models use the noisy channel model which allows them to break down the translation task into a translation model and a language model. There are 5 IBM models proposed in the literature[13]. IBM model 1 uses only lexical translation probabilities, IBM model 2 adds an absolute alignment model, IBM model 3 adds a fertility model, IBM model 4 uses a relative alignment model instead of absolute alignment model and IBM model 5 fixes a problem with a deficiency in the model.

2.1.2 Phrase-based SMT

Currently phrase-based models [138] form the basis of the best performing SMT systems, which translate small word sequences at a time instead of translating each word in isolation like word-based models. The context information can be incorporated into the translation model by learning the whole phrases instead of single words where a phrase is simply a sequence of words. The general idea of phrase-based translation is to segment the given source sentence into phrases, then translate each phrase and finally compose the target sentence from these phrase translations.

Phrase-based SMT does away with many of the problems associated with the original word-based formulation of statistical machine translation [13]. For instance, with multiword units less reordering needs to occur since local dependencies are frequently captured. For example, common adjective-noun alternations are memorized. However, since this linguistic information is not explicitly and generatively encoded in the model, unseen adjective-noun pairs may still be handled incorrectly. Since the focus of this thesis is Phrased-based Statistical machine translation from Bengali to English, a brief introduction of phrase based SMT is appropriate.

Motivation for Phrase-based Models

Word-based models for MT are based on translating individual word separately, however words may not be the best candidates for the smallest units of translation. Occasionally one word in the foreign language translates into two English words, or vice versa and word-based models often fail to work in such cases.

Figure 2.1 illustrates how the phrase-based model works for Bengali. The Bengali input sentence in segmented into so-called phrases (not necessarily linguistically motivated). Then each phrase is translated into an English phrase and then phrases are reordered if necessary. In the example, the five Bengali words and six English words are segmented into three phrase pairs. Since in English, the verb follows the subject, the English phrases have to be reordered.

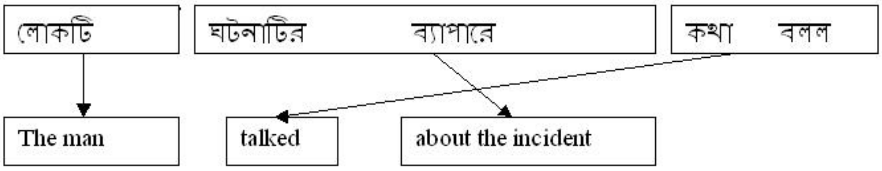


Figure 2.1: Phrase-based model

The Bengali word লোকটি best translates into the English phrase “the man”. This is best captured from a phrase translation table that maps not words but phrases. A phrase translation table of English translation for the Bengali word লোকটি may look as table 2.1.

Translation	Probability $p(e/f)$
the man	0.6
man	0.48
person	0.4
, man ,	0.2

Table 2.1: Phrase translation table

The current phrase-based models are not rooted in any deep linguistic notion of the phrase concept. So some phrases in the phrase-based model might have unusual grouping according to the syntactic theories. But since different languages have different grammatical rules, the context might sometimes provide useful clues to resolve translation ambiguities.

So phrase-based models are superior to word-based model due to several reasons. First, words are not always the best atomic units for translation due to frequent one-to-many mapping. Also translating word groups instead of single words helps resolve translation ambiguities. When large training corpora are available, a model can learn longer useful phrases and even sometimes memorize the translation of the entire sentence. Overall the model is conceptually much simpler compared to a word-based model.

Mathematical Definition

The context information can be incorporated into the translation model by learning the whole phrases instead of single words where a phrase is simply a sequence of words. The general idea of phrase-based translation is to segment the given source sentence into phrases, then translate each phrase and finally compose the target sentence from these phrase translations.

Phrase-based SMT does away with many of the problems associated with the original word-based formulation of SMT. For instance, with multiword units less reordering needs to occur since local dependencies are frequently captured. For example, common adjective-noun alternations are memorized. However, since this linguistic information is not explicitly and generatively encoded in the model, unseen adjective-noun pairs may still be handled incorrectly.

Current state-of-the-art MT systems use a phrase-based scoring model [68] for choosing among candidate translations in a target language. The phrase translation model is also based on the noisy channel mode. We use Bayes rule to reformulate the translation probability for translating a foreign sentence f into English as we saw in (2.2) which is repeated here as (2.4)

$$argmax_e p(e|f) = argmax_e p(f|e)p(e) \tag{2.4}$$

This allows for a language model $p(e)$ and a separate translation mode $p(f|e)$. During decoding, the foreign input sentence f is segmented into a sequence of I phrases f_1^I. We assume a uniform probability distribution over all possible segmentations. Each foreign phrase f_i in f_1^I is translated into an English phrase e_i in e_1^I. The English phrases may be reordered. Phrase translation is modeled by a probability distribution $\phi(f_i|ei)$. Due to Bayes rule, the translation direction is inverted from a modeling standpoint.

A relative distortion probability distribution $d(a_i – b_{i-1})$ is used to model the reordering of the English output phrases, where a_i denotes the start position of the foreign phrase which was translated into the ith English phrase, and b_{i-1} denotes the end position of the foreign phrase translated into the $(i-1)th$ English phrase. The distortion probability distribution $d(.)$ can be training using a simple distortion model $d(a_i – b_{i-1}) = \alpha_{|a_i – b_{i-1}-1|}$ with an appropriate value for the parameter α. The best English translation e_{best} for a foreign input f is defined as

$$e_{best} = argmax_e p(e|f) \tag{2.5}$$

$$= argmax_e p(f|e)p_{LM}(e) \tag{2.6}$$

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

$$p(f_1^I|e_1^I) = \prod_{i=1}^{I} \phi(f_i|e_i)d(a_i – b_{i-1}) \tag{2.7}$$

Now the phrase translation probability $\phi(f|e)$, reordering model d, and the language model $p_{LM}(e)$ can be multiplied together to form the phrase-based statistical machine translation model:

$$e_{best} = argmax_e \prod_{i=1}^{I} \phi(f_i|e_i)d(a_i–b_{i-1}) \prod_{i=1}^{|e|} p_{LM}(e_i|e_1...e_{i-1}) \tag{2.8}$$

It can be observed for the current model that the word match up between input and output could be quite good but the output might not be very good English. So more weight needs to be assigned to the language mode. And this can be achieved by introducing weights $\lambda\phi$ λd λLM that scale the contribution of each of the three components:

$$e_{best} = argmax_e \prod_{i=1}^{I} \phi(f_i|e_i)^{\lambda\phi} d(a_i–b_{i-1})^{\lambda d} \prod_{i=1}^{|e|} p_{LM}(e_i|e_1.......e_{i-1})^{\lambda LM} \tag{2.9}$$

Now we have a model structure, which is well known in the machine learning community: a log-linear model. Log-linear models have the following form:

$$p(x) = exp \sum_{i=1}^{n} \lambda_i h_i(x) \tag{2.10}$$

We can reformulate the phrase-based statistical machine translation model according to the log-linear model structure where the number of features $n = 3$ with feature functions $h_1 = log\phi$, $h_2 = logd$, $h_1 = logp_{LM}$.

$$p(e,a|f) = exp(\lambda_\phi \sum_{i=1}^{I} log\phi(f_i|e_i) + \lambda_d \sum_{i=1}^{I} logd(a_i–b_{i-1}) + \lambda_{LM} \sum_{i=1}^{|e|} logp_{LM}(e_i|e_1..e_{i-1}) \tag{2.11}$$

A log-linear model [97] directly models the posterior probability $P(e_1^I|f_1^J)$ using a log-linear combination of independent feature functions $h_1(.,.)...h_m(.,.)$ describing the relation of the source and target sentences, where λ is the model scaling factor.

The log-linear model (which contains the noisy channel model as a special case) is a generalization of the source-channel approach. It has the advantage that additional models or feature functions can be easily integrated into the overall system. Also the weighting of the different model components may lead to improvement in translation quality. So far the feature function we have discussed are language model probability, a phrase translation probability and distortion cost. Below we are going to discuss some other feature functions such as a reverse phrase translation probability, lexical translation probability, a reverse lexical translation probability, a word penalty and a phrase penalty. To set the weights λ minimum error rate training [95] is performed on the development set using BLEU[1] [98] as the objective function. The phrase translation probabilities were determined using maximum likelihood estimation over phrases induced from word-level alignments produced by performing GIZA++[2] training on each of the training corpora. The Pharaoh beam search decoder [62] is used to produce the translations after all of the model parameters have been set.

Reverse phrase translation Probabilities

According to Bayes rule we used an inversion of the conditioning of the translation probabilities: $p(e|f) = p(e)p(f|e)p(f)_{-1}$. However, in light of the phrase-based model we might have second thoughts on this. For example, in the training phase of learning the translation probabilities an uncommon foreign phrase f might mistakenly get mapped to a common English phrase e and $\phi(f|e)$ in this case is very high, even close to 1. Upon encountering the foreign phrase f again in the test data, this incorrect phrase translation will probably be used to produce the highest probability translation as translation model and language model both favours this translation. In such cases it would be beneficial to use the conditioning of phrase translation probabilities in the actual translation direction, $\phi(e|f)$. Also as feature functions it is possible to use the both translation directions, $\phi(e|f)$ and $\phi(f|e)$.

Lexicalized Weighting

The quality of a phrase translation pair can be validated by checking how well its words translate to each other. A lexical translation probability distribution $w(f|e)$ is needed for that. It can be estimated by relative frequency from the same word alignments as the phrase model.

$$w(f|e) = \frac{count(f, e)}{\sum_{f^{‘}} count(f^{‘}, e)} \tag{2.12}$$

[1] BLEU is an IBM-developed metric which measures how close a candidate translation is to a reference translation by doing an n-gram comparison between both translations. It will be discussed in detail in section 4.4.

[2] GIZA++ is a tool that performs alignment between two parallel aligned corpora

A special English NULL token is added to each English sentence and aligned to each unaligned foreign word. Given a phrase pair $\overline{f}, \overline{e}$ and a word alignment a between the foreign word positions $i = 1, ...n$ and the English word positions $j = 0, 1, ...m$, we compute the lexical translation probability $lex(\overline{f}|\overline{e}, a)$ by:

$$lex(\overline{f}|\overline{e}, a) = \prod_{i=1}^{n} \frac{1}{|j|(i,j) \in a|} \sum_{\forall (i,j) \in a} w(f_i|e_j) \tag{2.13}$$

If there are multiple alignments a for a phrase pair, the one with the highest lexical weight is selected:

$$lex(\overline{f}|\overline{e}) = max_a lex(\overline{f}|\overline{e}, a) \tag{2.14}$$

During translation, the lexical translation probability $lex(\overline{f}|\overline{e})$ can be used as an additional feature function. So for the phrase-based model, $p(f|e)$ can be extended to :

$$p(f_1^I|e_1^I) = \prod_{i=1}^{I} \phi(f_i|e_i) d(a_i – b_{i-1}) lex(f_i|e_i, a)^{\lambda} \tag{2.15}$$

The parameter λ defines the weight of the lexical translation probability $lex(\overline{f}|\overline{e})$ and usually good values for the parameter are around 0.25. Also as mentioned in the previous section it might be useful to have both translation directions in the model which are, lexical translation probability $lex(\overline{f}|\overline{e}, a)$ and reversed lexical translation probability $lex(\overline{e}|\overline{f}, a)$

For example, in the figure 2.2, the alignment is shown between the English phrase ‘‘*did not destroy*” and the Bengali phrase নষ্ট করে -nosto kore . The lexical weight for this phrase pair is the product of three factors, one for each English word. The English word ”*not*” is aligned to the Bengali word করেনি -korene, so the factor is $w(not|$ করেনি $)$. The English word ”*did*” is not aligned to any foreign word, so the factor is $w(did|NULL)$ and the English word destroy is aligned to two Bengali words ক্ষয় ক্ষতি -khoikhoti, so the factor is the average of the two corresponding words' translation probability.

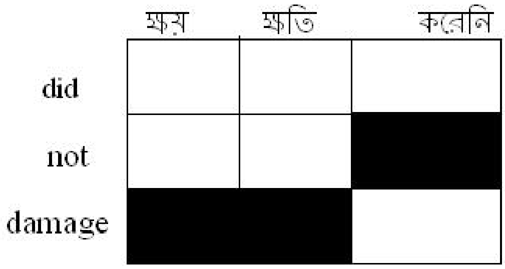


Figure 2.2: Lexicalized weighting

$$lex(\overline{f}|\overline{e}, a) = lex(f_1, f_2, f_3|e_1, e_2, e_3, a)$$

$$= w(did|NULL) \text{ x } w(not|\text{ করেনি }) \text{ x } \tfrac{1}{2}(w(damage|\text{ ক্ষয় }) + w(damage|\text{ ক্ষতি }))$$

Word penalty

In the phrase-based model, we haven't yet explicitly modelled the output length in terms of number of words. However, the language model prefers shorter translations. A word penalty has been introduced which adds a factor w for each produced word for too short or too long output. If $w < 1$ the scores of shorter translations are improved and if $w > 1$ longer translations are preferred. This parameter is very effective in tuning output length and sometimes helps to improve translation quality.

Phrase penalty

Any phrase translation has to be segmented into foreign phrases before it can be applied to a new input sentence. In the phrase-based model we haven't yet explicitly modelled this segmentation. A phrase penalty has been introduced which adds a factor ρ for longer phrases or shorter phrases. If $\rho < 1$ longer phrases are preferred and if $\rho > 1$ shorter phrases are preferred.

2.1.3 Factored Translation Models

Current state-of-the-art phrase-based SMT systems are limited to the mapping of small text phrases without any explicit use of linguistic information such as morphological, syntactic or semantic information. Addition of such information as a preprocessing or post-processing step has demonstrated to be valuable in SMT. Much of the translation process is best explained with morphological, syntactic, semantic, or other information that is not typically present in parallel corpora. Factored translation models [65] incorporate such information with the training data to build richer models of translation. There are several reasons for integration of linguistic information into the translation model: a) the translation model can operate on a more general representation such as lemmas instead of surface form of words and thus can draw on richer statistics to overcome the data sparseness problems due to limited training data; b) many aspects of translation can be best explained on a morphological, syntactic, or semantic level and translation model having access to such information allows direct modelling of these aspects.

The basic idea behind factored translation models is to represent phrases not simply as sequences of fully inflected words, but instead as sequences containing multiple levels of information. A word in this framework is not a single token, but a vector of factors that represent different levels of annotation. This enables straight-forward integration of part-of-speech tags, morphological information, and even shallow syntax. Instead of dealing with linguistic markup in preprocessing or post processing steps we build a system that integrates this information into the decoding process to better guide the search.

The use of factors introduces several advantages over current phrase-based approaches:

- Better handling of morphology by translating in multiple steps.
- Better decisions can be facilitated by linguistic context when selecting among translations.
- Provides many new modelling possibilities due to linguistic mark up of the training data.

The example presented below demonstrates the limitation of the traditional surface word approach in SMT in handling morphology. In phrase-based SMT, each word form is treated as a token in itself as a result the translation model treats, for example, the word "report" completely independent of the word "reports". Any instance of "report" in the training data does not add any knowledge to the translation of "reports". So while the translation of "report" may be known to the model, the word "reports" may be unknown and the system will not be able to translate it. Although this problem does not show up as strongly in English, - it does constitute a significant problem for morphologically rich languages such as Bengali, Arabic, German, Czech, etc which (unlike English) have a rich inflectional morphology.

Factored translation models translate lemmas and morphological information separately, and combine this information on the output side to ultimately generate the output surface words. Below figure[3] 2.3 illustrates one such model where morphological analysis and generation are decomposed into three mapping steps which are translation of lemmas, translation of part-of-speech and morphological information, and generation of surface forms.

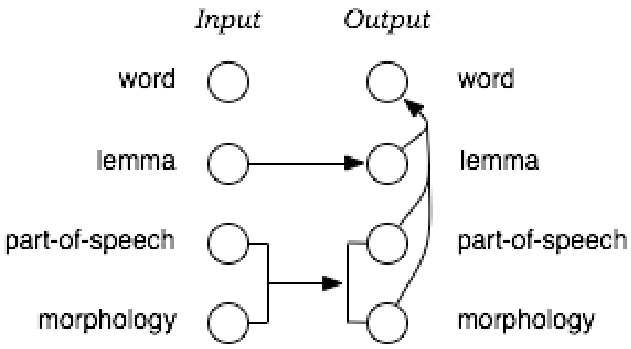


Figure 2.3: Factored model

2.2 Syntax-based MT

One of the key limitations of phrase-based SMT systems ([82], [68]) is that they use little or no syntactic information. Sometimes syntactic information is crucial in accurately modeling many phenomena during translation as different languages differ in their syntactic structure. Currently SMT systems which incorporate syntactic information have received a great deal of interest. One of the advantages of syntax-based SMT is that they enforce syntax motivated constraints in translation and capturing long-distance/non-contiguous dependencies.

Some approaches have used syntax at the core ([132], [6], [135], [44], [36], [49], [85]) while others have integrated syntax into existing phrase-based frameworks ([133], [23], [26], [101]).

Xia and McCord [133] use syntactic knowledge in their approach. They use pattern learning in their reordering system. In their work they parse and align sentences in the training phase and derive reordering patterns. From the English-French Canadian Hansard [4] they extract 56,000 different transformations for translation. In the decoding phase they use

[3] Figure taken from http://www.statmt.org/moses/?n=Moses.FactoredModels

[4] LDC Catalog No.: LDC95T20

these transformations on the source language. The main focus then is monotonic decoding.

Quirk et al. [101] used a dependency parser to learn certain translation phrases, in their work on 'treelets'. Marcu et al. [81] present a syntax-based approach with phrases that achieves a convincing quality improvement over phrases without these syntax rules.

Several researchers have proposed models where the translation process involves syntactic representations of the source and/or target languages. Some models use bitext grammars which simultaneously parse both the source and target languages and others use syntactic information in the target language alone. Based on the kind of linguistic information which is made use of, syntactic SMT can be divided into four types: tree-to-string, string-to-tree, tree-to-tree, and hierarchical phrase-based.

The **tree-to-string approach** ([26], [91], [78], [77]) supposes that the syntax of the source language is known. This approach can be applied when a source language parser is available.

Syntactically motivated rules based on clause restructuring are used in reordering models. Collins et al. [26] describe reordering based on a dependency parse of the source sentence. In their approach they have defined six hand-written rules for reordering German sentences. The reordering rules however are language-pair (German-English) specific and hand-written. In brief, German sentences typically have the tensed verb in second position; infinitives, participles and separable verb particles occur at the end of the sentence. These six reordering rules are applied sequentially to the German sentence, which is their source language. Three of their rules reorder verbs in the German language, and one rule reorders verb particles. The other two rules reorder the subject and put the German word used in negation in a more English-like position. All their rules are designed to match English word ordering as much as possible. Their approach shows that adding knowledge about syntactic structure can significantly improve the performance of an existing state-of-the-art SMT system.

Nguyen and Shimazu [91] presented a more general method in which lexicalized syntactic reordering models based on Probabilistic Context-free Grammars(PCFGs) can be learned from source-parsed bitext and then applied in the preprocessing phase. Liu et al. [78] changed the translation unit from phrases to tree-to-string alignment templates (TATs). TATs were represented as xRs rules. In order to overcome the limitation that TATs can not capture non-constituent phrasal translations, Liu et al. [77] proposed forest-to-string rules.

The **string-to-tree approach** ([45], [43]) focuses on syntactic modelling of the target language in cases where there are syntactic resources such as treebanks and parsers.

Yamada and Knight [135] use a parser in the target language to train probabilities on a set of operations that transform a target parse tree into a source string. Graehl and Knight [45] proposed the use of target tree- to-source-string transducers (xRS) to model translation. In xRS rules, the right-hand-side(rhs) of the target side is a tree with non-terminals(NTs), while the rhs of the source side is a string with NTs. Galley et al. [43] extended this string-to-tree model by using Context-Free parse trees to represent the target side. A tree could represent multi-level transfer rules.

A **tree-to-tree model** [44] translation makes use of a syntactic tree for both the source and target language. Like in the tree-to-string model, a set of operations apply, each with some probability, to transform one tree into another. However, when training the model, trees for both the source and target languages are provided.

Hajič et al. [48] proposed a Tree-to-tree alignment technique known as probabilistic tree substitution grammars which can be trained on parse trees from parallel treebanks. Gildea [44] also proposed tree-based probabilistic alignment methods. These methods reorder, insert or delete sub-trees of one side to reproduce the other side. The method aligns non-isomorphic phrase-structure trees using a stochastic tree substitution grammar (STSG). This approach involves the altering of the tree structure in order to impose isomorphism, which impacts on its portability to other domains.

The **hierarchical phrase-based approach** [23] constrains phrases under context-free grammar structure without any requirement of linguistic annotation. Chiang [23] presents a hierarchical phrase based model that uses hierarchical phrase pairs, which are formally productions of a weighted synchronous context-free grammars.

Reranking methods ([66], [96], [113]) have also been proposed which make use of syntactic information. In these methods a baseline system is used to generate N-best output. Syntactic features are then used in a second model that reranks the N-best lists, in an attempt to improve over the baseline approach.

2.3 Summary

In this chapter, we reviewed the background knowledge about MT needed to follow the rest of the thesis. We mainly discussed the Phrase-based SMT, factored translation models and syntax-based MT. After highlighting the limitations of word based models, we considered phrase-based models that translate small word sequences at a time instead of translating each word in isolation like word-based models. We then saw how factored models could deal with morphological information and finally how syntax could also be taken into account.

Chapter 3

Bengali Language and Bengali Dataset

Since the focus of this thesis is Statistical machine translation from Bengali to English, a brief introduction to the Bengali language is appropriate. This chapter will provide an introduction to the Bengali writing style, alphabet and some peculiarities; no previous knowledge of Bengali grammar is required.

3.1 The Bengali Alphabet

The Bengali writing system unlike the Latin script is not a purely alphabetic script. However, it is a variant of Eastern Nagari script used throughout Bangladesh and eastern India including Assam, West Bengal and the Mithila region of Bihar, known as an abugida called the Bengali script. The Eastern Nagari script is believed to have evolved from a modified Brahmic script and is similar to the Devanagari abugida used for Sanskrit and many modern Indic languages such as Hindi. The Bengali script has close historical relationships with the Assamese script, the Oriya script and Mithilakshar which is the native script for the Maithili language.

The Bengali script is a cursive script. It contains eleven signs denoting the independent form of nine vowels and two diphthongs, and thirty-nine signs denoting the consonants with "inherent" vowels. The concept of capitalization is absent in the Bengali orthography. There is no variation in initial, medial and final forms of letters as in the Arabic script. The letters run from left to right on a horizontal line, and spaces are used to separate orthographic words. The table 3.1 and 3.2 shows all the vowels and consonants.

Letter	Transliteration	Letter	Transliteration
অ	a	ঋ	*ri*
আ	$\overline{a}$	এ	*e*
ই	*i*	ঐ	*ai*
ঈ	$\overline{i}$	ও	*o*
উ	*u*	ঔ	*au*
ঊ	$\overline{u}$		

Table 3.1: Bengali vowels

Letter	Transliteration	Letter	Transliteration	Letter	Transliteration
ক	*k*	ঢ	$\overline{dh}$	র	*r*
খ	*kh*	ণ	*n*	ল	*l*
গ	*g*	ত	*t*	শ	*s*
ঘ	*gh*	থ	*th*	ষ	*sh*
ঙ	$\underline{n}$	দ	*d*	স	*sh*
চ	*ch*	ধ	*dh*	হ	*h*
ছ	*chh*	ন	*n*	ড়	$\underline{r}$
জ	*j*	প	*p*	ঢ়	$\underline{rh}$
ঝ	*jhi*	ফ	*ph*	য়	*e*
ঞ	*n*	ব	*b*	ৎ	*ta*
ট	$\underline{t}$	ভ	*bh*	◌ং	*ang*
ঠ	$\underline{th}$	ম	*m*	◌ঃ	*ah*
ড	$\overline{d}$	য	*j*	◌ঁ	*u*

Table 3.2: Bengali consonants

3.2 Bengali Grammar

3.2.1 Word order

As a head-final language, Bengali follows Subject Object Verb(SOV) word order. Unlike the prepositions used in English and other European languages, Bengali makes use of postpositions. Also in Bengali determiners follow the noun, while numerals, adjectives, and possessors precede the noun.

The basic word order does not need to be changed in the yes-no question in Bengali; instead, the low (L) tone of the final syllable in the utterance is replaced with a falling (HL) tone. In a yes-no question, additionally optional particles for example, কি -ki (what), না -na (no) are often encliticized onto the first or last word. By fronting the wh-word to focus position, which is typically the first or second word in the utterance, wh-questions are formed.

3.2.2 Nouns

Nouns and pronouns are inflected for case, including nominative, objective, genitive (possessive), and locative. The case marking pattern for each noun being inflected depends on the noun's degree of animacy. When a definite article such as -টা -ta (singular) or -গুলা -gula (plural) is added, nouns are also inflected for number.

As in many East Asian languages (e.g. Chinese, Japanese, Thai, etc.), nouns in Bengali cannot be counted by adding the numeral directly adjacent to the noun. The noun's measure word (MW) must be used between the numeral and the noun. Most nouns take the generic measure word -টা -ta, though other measure words indicate semantic classes for example -জন -jon (for humans).

Measuring nouns in Bengali without their corresponding measure words would typically be considered ungrammatical. For example:

আট বিড়াল -at biṛal (eight cats) instead of আটটা বিড়াল at-ta biṛal (eight cats) (3.1)

However, when the semantic class of the noun is understood from the measure word, the noun is often omitted and only the measure word is used. For example:

শুধু একজন থাকবে। -Shudhu ekjon thakbe (Only one will remain) (3.2)

This would be understood to mean “Only one person will remain.”, given the semantic class implicit in -জন -jon. In this sense, all nouns in Bengali, unlike most other Indo-European languages, are similar to mass nouns.

3.2.3 Verbs

Verbs divide into two classes: finite and non-finite. Non-finite verbs have no inflection for tense or person, while finite verbs are fully inflected for person (first, second, third), tense (present, past, future), aspect (simple, perfect, progressive), and honor (intimate, familiar, and formal), but not for number. Conditional, imperative, and other special inflections for mood can replace the tense and aspect suffixes. The number of inflections on many verb roots can total more than 200.

Inflectional suffixes in the morphology of Bengali vary from region to region, along with minor differences in syntax. Bengali differs from most Indo-Aryan Languages in the zero copula, where the copula or connective be is often missing in the present tense. For example:

He is a teacher. সে শিক্ষক -Shay Shikkhok (he teacher). (3.3)

In this respect, Bengali is similar to Russian and Hungarian.

3.2.4 Preposition

In Bengali, there is no concept of preposition. English prepositions are handled in Bengali using inflections on the referenced objects and/or by post-positional words after the objects. Inflections get attached to the reference objects.

There are a few inflections in Bengali as described in table 3.3:

Bengali inflections	Bengali inflections
Φ -null	-কে -ke
-◌ে -e	-রে -re
-য় -y	-◌েরে -ere
-য়ে -ye	-র -r
-তে -te	-◌ের -er
-◌েতে -ete	

Table 3.3: Bengali inflections

The placeholder indicated by a dashed circle represents a consonant or a conjunct. For example, if -ে◌ inflection is attached to the word কাগজ -kagoj (newspaper) the inflected word is কাগজে -kagojr-e (in newspaper). On the other hand, post-positional words are independent words. They have meanings of their own and are used independently like other words. A post-positional word is positioned after an inflected noun (the reference object). Some examples of the post positional words in (colloquial) Bengali are:

Bengali post positional word
দিয়ে -diye (by)
থেকে -theke (from)
জন্য -jonno (for)
কাছে -kachhe (near)
সামনে -samne (in front of)

Table 3.4: Bengali post-positional word

3.2.5 Compound Words

There are four principal divisions of compound word, which are nouns, adjectives, verbs and adverbs. Below we describe each of them:

Compound Nouns

There are in Bengali two kinds of compound nouns. The first one is formed by stringing together two or more nouns, omitting the conjunctions and inflecting only the final one. For example, পিতামাতা -pitamata (father and mother), মাংসরক্ত -manshorokto (flesh and blood).

The second one is formed by prefixing to the final noun words of almost any description. For example by prefixing another noun ধর্মপুস্তক (the holy book), by prefixing an adjective to the noun ভালমনুষ্য -valomannuso (a good man).

Compound Adjectives

The compound adjectives, like the substantives, are of two kinds. The first kind admits various combinations. For example by uniting two adjectives together মহাজাতীয় -mohajatio (of an excellent race), by uniting an adjective with a substantive and shorting the final হতবুদ্ধি -hotobuddhe (having lost his senses) .

The second kind is formed by the union of a noun with a verbal adjective or past participle. For example, আনন্দদায়ক -anondodaiok (joy-giving) হস্তগত -hostogoto (come to hand).

Compound Verb

There are several kinds of compound verbs, formed principally by combining a noun or participle with a verb. When thus formed, the compound is conjugated as a simple verb, The following are some of the principle compounds:

Nominals are formed by uniting any noun or past participle with an auxiliary verb such as বিক্রয়করণ -bikroikoron (to sell),গমনকরণ -gomonkoron (to go).

Double compounds of this sort are formed by the union of two or more nouns with a verb such as ভোজনপানকরন -vhozonpankoron (to eat and drink).

Transformatives are formed by a participle with the verb যাওন -jhaon (going), and signify the becoming of what is expressed by the participle. For example, উঠিয়াযাওন uthiejhaon (going up).

Compound Adverbs

Compound adverbs are formed by prefixing some indeclinable word to a noun such as যাবৎ-জীবন -jabotjinob (as long as life lasts), যথাশক্তি -jothashokti (to the extend of one's power). Compounds with রূপে, -rupa (in an) মতে -motha(by) as their final member may be considered as adverbs. For example বিলক্ষনরূপে -belockhonrupa (in an excellent way or form).

3.3 Bengali Dataset

MT from Bengali to English has become one of the most vital tasks for Natural Language Processing in the Bengali language [32]. Bengali, an Indo-Aryan language, is the native language of people of Bangladesh which has more than 200 million native speakers around the world. It is the seventh most spoken language in the world, second in India. Although being among the top ten most widely spoken languages around the world, the Bengali language still lacks significant research in the area of natural language processing specifically in MT and also lacks resources. Below we describe all the datasets used in our experiments for Bengali-English SMT.

3.3.1 Bilingual Corpora for SMT

The corpus we used for training the system was provided by the Linguistic Data Consortium[1] (LDC) containing around 11,000 sentences. It contains newswire text from the BBC Asian Network and some other South Asian news websites. A bilingual Bengali-English dictionary collected from different websites was also used as part of the training set which contains around 55K words. For our language model we used data from the English section of EuroParl combined with the LDC training set. The development set used to optimize the model weights in the decoder, and the test set used for evaluation were taken from the same LDC corpus mentioned above. The corpus statistics are shown in the table 3.5.

resources	used for	sentences
LCD Training Set(Bengali-English)	Phrase table + LM	11226
Dictionary(Bengali-English)	Phrase table	55312 (words)
Europarl(English)	LM	182234
LCD Development Set(Bengali-English)	Development	600
LDC Test Set (1 ref. Bengali-English)	Test	1000
Extended Test Set (3 ref. Bengali-English)	Test	1000

Table 3.5: Dataset statistics

3.3.2 Monolingual Corpora for SMT

We have a large monolingual Bengali dataset which contains more than one million sentences. The monolingual corpus was provided by the Center for Research on Bangla Language Processing, BRAC University, Bangladesh. The corpus was built by collecting text from the Prothom Alo newspaper website and contains all the news available for the year of 2005 (from 1st January to 31st December) - including magazines and periodicals. There are 18,067,470 word tokens and 386,639 distinct word types in this corpus.

3.3.3 Extended Test Set for SMT

The test set provided by the LDC contains only single reference translations between Bengali and English. Unavailability of multiple reference translations can have impact on the BLEU

[1] LDC Catalog No.: LDC2008E29.

score. Additionally having just one reference translation does not bring of the full potential of translation quality for a MT system. So we extended the LDC test set by adding two new English reference translation sets. There are 1000 sentences in the LDC test set. We created two new reference English test sets by translating these 1000 sentences. Two native Bengali speakers were involved in the translation.

3.3.4 Test Set for Compound Words and Transliteration Module

We created a test set for the compound word splitting task. We collected 280 compound words from the Bengali Prothom-Alo monolingual corpora manually for evaluating our compound word splitting approach.

In order to evaluate the performance of our of stand alone transliteration module, we also created a development test set and a blind test set of 220 and 210 name pairs respectively between Bengali-English. The test sets were manually created from a bilingual corpus between Bengali-English.

3.4 Summary

In this chapter, we reviewed the background knowledge about the Bengali language including Bengali writing style, alphabet and some Bengali grammar. We also discussed the Bengali dataset we used for all our experiments in this thesis.

Chapter 4

Bengali Dependent Approaches

In this chapter we describe the background and approaches for Bengali dependent processes such as transliteration, compound word processing, preposition handling and word reordering.

4.1 Transliteration

Transcribing the words from a source language orthography into the target language orthography is called transliteration. The transliterating of names(specially, named entities) independent of end-to-end MT has received a significant amount of research, e.g., ([61], [20], [5]). Named entities (NEs) are noun phrases in the sentence that refer to persons, locations and organizations.

The task of transliteration can be classified into two categories or can be a hybrid of these two categories: 1) Transliteration generation and 2) Transliteration discovery.

4.1.1 Transliteration Generation

Transliteration generation usually uses generative approaches to create the equivalent transliteration from the source language to the target language. In this approach, generally either the pronunciation of named entities or their spelling or a combination of both are used to train a corpus and generate the output. Usually these transliteration methods are useful for MT and cross lingual information retrieval tasks.

There has not been any significant work done in literature on using the generative approach for automatic transliteration of proper nouns from Bengali to English. UzZaman et al. [129] introduced a comprehensive English-to-Bengali transliteration scheme that handles the full complexity of the Bengali script with the assistance of a phonetic lexicon. In their transliteration scheme, they used two types of mapping: a direct phonetic mapping and a lexicon-enabled phonetic mapping to transliterate source to the goal language script (Bengali). Recently, Google labs India [1] introduced a transliteration system that converts from English Roman characters to Bengali characters. The system allows the user to type Bengali words phonetically in English script and still have them appear in their correct alphabet.

However, generative approaches for transliteration have been used for other languages. Arbabi et al. [7] presented an algorithm for transliteration from Arabic to English. In their approach, diacritization(the task that involves adding diacritics such as, short vowels, or special markers to the standard written form) is performed on Arabic names by inserting appropriate short vowels into the words which otherwise lack them. Then using a parser and table lookup, the vowelized Arabic names are converted into a phonetic Roman representation. Finally using this phonetic representation and table lookup, the correct spelling in the target language is produced.

Knight and Greehl [61] describe a pronunciation-based approach to transliterate from Japanese to English. They build five probability distributions in their adopted generative story of converting an English name into Japanese: $P(w)$ is used to generate written English sequences, $P(e|w)$ pronounces English word sequences; $P(j|e)$ converts English sounds into Japanese sounds, $P(k|j)$ converts Japanese sounds to katakana writing and $P(o|k)$ introduces misspellings caused by optical character recognition(OCR). They evaluated the performance of their approach on names from Japanese newspaper articles which showed an accuracy of 64 percent compared to human evaluation which was 27 percent.

Based on the source-channel framework, Stalls and Knight [120] described an Arabic to English back-transliterate system where the transliteration approach is based on a generative model of how an English name is transliterated into Arabic. Al-Onaizan and Knight [5] presented a spelling-base model from Arabic to English which is based on Stalls and Knight's phonetic-based model. Their spelling-based model directly maps English letter sequences into Arabic letter sequences with a probability $P(a|w)$. They evaluated the performance of

[1] http://www.google.com/transliterate/indic/Bengali

their approach on several settings: one phonetic-based model alone, spelling-based model alone and finally both models combined.

AbdulJaleel and Larkey [1] describe a statistical transliteration system from English to Arabic in the context of Cross Lingual Information Retrieval (CLIR). They present a simple statistical technique called selected n-gram modelling to train an English to Arabic transliteration model from pairs of names. The selected n-gram model has a two-stage training procedure: it first learns which n-gram segments should be added to the unigram inventory for the source language, and then a second stage learns the translation model over this inventory. This technique requires no heuristics or linguistic knowledge of either language. They evaluated the performance of their statistically-trained model and a simpler hand-crafted model on a test set of named entities from the Arabic AFP corpus and demonstrated that their approach outperforms two online translation sources.

Sherif and Kondrak [114] proposed a language-independent bootstrapping approach for training a stochastic transducer which learns scoring parameters from an English-Arabic bitext for the task of extracting transliterations. They showed that their bootstrapping transducer performs as well or better than an Arabic-English specific similarity metric on the task of Arabic-English transliteration extraction. Sherif and Kondrak [115] also proposed two substring based transliteration approaches: a dynamic programming algorithm, and a finite-state transducer based on phrase-based models of MT for modeling transliteration. They showed that their substring-based transducer approach outperforms a state-of-the-art letter based approach.

Also a novel spelling-based method for the automatic transliteration of proper nouns from Arabic to English in the context of MT was proposed by Kashani et al. [58]. They exploit various types of letter-based alignments. Their approach consists of three phases: the first phase uses single letter alignments, the second phase uses alignments over groups of letters to deal with diacritics and missing vowels in the English output, and the third phase exploits various knowledge sources to repair any remaining errors. Their results show a top-20 accuracy rate of 88 % and 86 % on development and blind test sets respectively.

4.1.2 Transliteration Discovery

Transliteration discovery methods mostly rely on the structural similarity between languages and writing systems. These methods usually use parallel corpora and some distance metrics. Generally these methods are used in order to build bilingual named-entity lexicons.

Some research has been done in discovering the named entity equivalents in comparable and parallel corpora. The common approach is to have a simple transliteration module and use some temporal and spatial clues in the corpora to confirm or reject the candidates as possible equivalent pairs.

Samy et al. [110] describe a transliteration discovery approach which uses an Arabic-Spanish parallel corpora and a Spanish named entity tagger to tag Arabic named entities. For each sentence pair aligned together, they use a simple mapping scheme to transliterate all the words in the Arabic sentence and return those matching with NEs in the Spanish sentence as the NEs in Arabic. They reported the precision and recall values of 90 % and 97.5 % respectively.

Freeman et al. [41] presented an approach for encoding language knowledge directly into their Arabic-English fuzzy matching algorithm for the task of transliteration detection and extraction. They achieved significant improvement in F-score in the context of cross linguistic name matching in English and Arabic by augmenting the classic Levenshtein edit-distance algorithm with character equivalence classes.

Sproat et al. [119] presented an approach for transliteration between Chinese and English using comparable corpora (corpora where texts in the two languages deal in some of the same topics and share references to named entities but are not translations of each other) with tagged NEs in both Chinese and English languages. They presented two distinct methods for transliteration, one approach using phonetic transliteration, and the second using the temporal distribution of candidate pairs. The combination of the approaches achieves better results. They also propose a novel score propagation method that utilizes the co-occurrence of transliteration pairs within document pairs. This propagation method achieves further improvement over previous approaches.

4.1.3 Challenges in Bengali SMT

In our Bengali to English SMT system a lot of words are improperly translated or ignored during the translation phases due to the limited amount of parallel data. Analysis of the output of our SMT system shows that most of these words are NEs which need to be transliterated. For example the table 4.1 illustrates the output of the SMT system.

Source sentence:

উওর$_1$ পশ্চিমের$_2$ বাগাদো$_3$ এবং$_4$ দাবেইবা$_5$ শহরে$_6$ সংঘর্ষে$_7$ কমপক্ষে$_8$ ৫৪$_9$ জন$_{10}$
সৈনিক$_{11}$ এবং$_{12}$ ৫০$_{13}$ জন$_{14}$ গেরিলা$_{15}$ নিহত$_{16}$ হয়েছেন$_{17}$

Utter$_1$ Poschim$_2$ Bagado$_3$ ebong$_4$ Debeiba$_5$ shohore$_6$ shonghorsha$_7$ kompokha$_8$ 54$_9$ jon$_{10}$ shoino$_{11}$ ebong$_{12}$ 50$_{13}$ jon$_{14}$ guerilla$_{15}$ nihoto$_{16}$ hoecha$_{17}$.

North$_1$ western$_2$ Bagado$_3$ and$_4$ Debeiba$_5$ cities$_6$ in-clashes$_7$ at-least$_8$ 54$_9$ soldiers$_{11}$ and$_{12}$ 50$_{13}$ guerillas$_{15}$ have$_{16}$ died$_{17}$.

SMT output:

North west বাগাদো and in দাবেইবা in at least ৫৪ soldiers and 50 philippine guerrillas have been killed.

Reference sentence:

At least 54 soldiers and 50 guerillas have died in clashes around the northwestern cities of Bagado and Dabeiba.

Table 4.1: SMT output

In table 4.1, both city names বাগাদো and দাবেইবা were not translated into the target language. As can be seen in the reference sentence, transliteration of these two names would clearly improve the readability of the text. So we want to investigate how the introduction of a transliteration module to our SMT system affects translation accuracy.

4.1.4 Our Transliteration Module

As described in the previous section there are various approaches to transliteration and one of the main deciding factors on which approach to use is the application in which the transliteration module would be integrated. For CLIR, the transliteration discovery methods using the parallel corpora and comparable corpora are applicable however generative methods are more appropriate. For building a bilingual named entity dictionary and discovering transliterations, the parallel and comparable corpora methods are suitable. And in the context of machine translation, generative approaches are the most appropriate.

We used a generative approach for transliteration where we generate an equivalent transliteration from the source language to the target language. The transliteration module was inspired by the work done by AbdulJaleel and Larkey [1] for their statistical transliteration system from English to Arabic in the context of Cross Lingual Information Retrieval

(CLIR).

We discuss a transliteration approach which is mostly language independent and can be added to any SMT system. The only resource needed is a parallel name list between two languages. We describe the transliteration technique as an add-on module for a Bengali-English SMT system.

Our transliteration module looks for the best candidates for transliteration of a given Bengali named entity using the Viterbi algorithm in a Hidden Markov Model (HMM) framework. Our approach treats each letter and each word as word and sentence respectively. The transliteration module is trained using GIZA++[2] on a parallel name list of about 580000 names collected from the West Bengal election website[3]. The language model is built using the SRILM toolkit over a list of names in English collected from the US census bureau[4] and west Bengal election website.

Given the output of our SMT system with untranslated words as seen in table 4.1, the transliteration module first identifies all the untranslated named entities that need to be transliterated. First we do a lookup of the untranslated Bengali word in a bilingual Bengali-English lexicon in order to resolve transliteration as a pre-processing step. A further lookup of the untranslated word is done in the lexicon after applying a Bengali morphological analyser in order to remove inflection from the word.

Then the transliteration module is applied to find the best English matches for all the explicitly written letters of named entities in Bengali. The generated transliterations are checked against an English monolingual dictionary containing 94646 names from the US census bureau and other named entities collected from web resources to see if they have close string distances using the Levenshtein distance [5] to some entities in the dictionary or if they might appear in the dictionary. If the generated transliteration appears in the dictionary, the transliteration is assumed to be correct.

For further comparison with entries in the dictionary, vowels are stripped-off from both the candidate and the dictionary entry looking for a match. If none of the dictionary entries are found to match then the generated transliteration is left as is. Then we replace the

[2] GIZA++ is a tool that performs alignment between two parallel aligned corpora

[3] http://www.indian-elections.com/assembly-elections/west-bengal/

[4] http://www.census.gov/

[5] It is a metric for measuring the amount of difference between two sequences. The Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character.

untranslated words with these transliterations in the output of our SMT system. Figure 4.1 below illustrates the architecture of the transliteration module in our system.

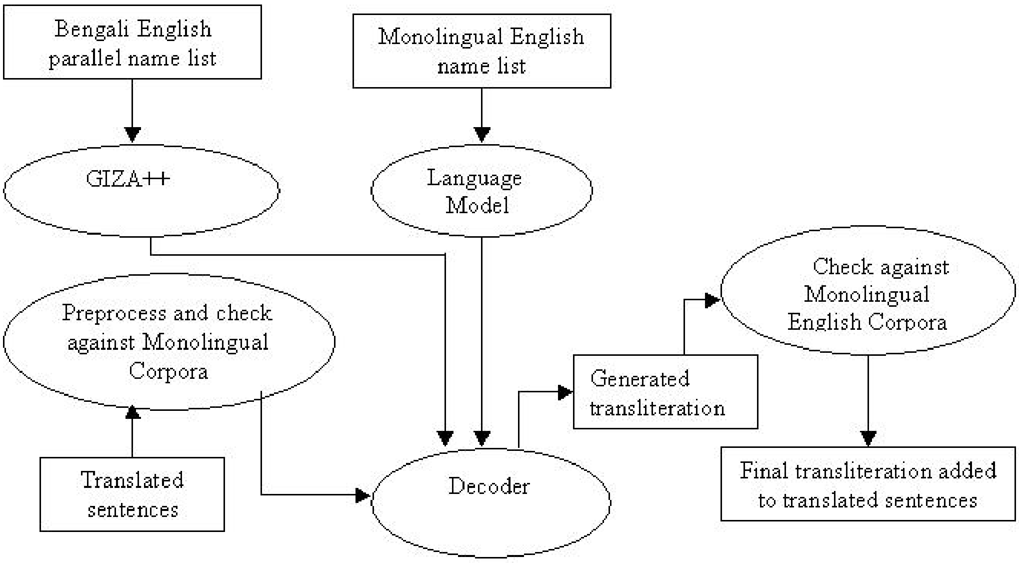

Bengali English parallel name list
Monolingual English name list
GIZA++
Language Model
Check against Monolingual English Corpora
Preprocess and check against Monolingual Corpora
Generated transliteration
Translated sentences
Decoder
Final transliteration added to translated sentences

Figure 4.1: Transliteration system

The transliteration module functions as described in Algorithm 1.

Algorithm 1 Transliteration module

```
Given output of SMT system in English containing Bengali words
Identify all the Bengali named entities
for Each word in the Bengali named entities do
  Lookup word in a bilingual dictionary
  if The Bengali word matches some dictionary entry then
    propose English part of the dictionary entry as final transliteration
  else
    Remove inflection from word using Morphological analyser
    Lookup uninflected word in bilingual dictionary
    if Uninflected word matches some dictionary entry then
      propose the English part of the dictionary entry as final transliteration
    else
      Generate transliterated word from Bengali word
      Check transliteration against a Monolingual English Dictionary
      if English transliteration matches some dictionary entry then
        propose English part of the dictionary entry as final transliteration
      else
        remove all vowels from English transliteration and compare stripped English
        transliteration with stripped words from bilingual dictionary using Levenshtein
        distance
        if transliteration matches with some dictionary entry then
          propose the dictionary entry as final transliteration
        else
          propose the generated transliteration as final transliteration
        end if
      end if
    end if
  end if
end for
```

For example, let us consider that the Bengali word কানাডাতে -canadate (Canada) needs to be transliterated by the transliteration module. First we check against a bilingual Bengali-English lexicon to see if the word কানাডাতে can be found there. If we find the word, we propose the corresponding English entry as correct transliteration. Otherwise, we apply Bengali morphological analyser on the word কানাডাতে and after removing inflection we have কানাডা. Then we check again against the bilingual Bengali-English lexicon to see if the word কানাডা can be found there. If we find the word, we propose the corresponding English entry which would be "Canada" as the correct transliteration.

Otherwise, we generate a transliteration based on our transliteration model, which for কানাডা would be "Kanadha". Then we do string comparison to check the generated word "Kanadha" against the monolingual English dictionary using Levenshtein distance. If no matches are found, we perform further string comparison with the monolingual English dictionary by stripping off the vowels from both the word and the dictionary entries. If a match is found we propose that as the correct transliteration which would be "Canada" for this example. Otherwise "Kanadha" is proposed as the resolved transliteration.

Note that we added the transliteration module as a post-processing step instead of pre-processing. Adding a transliteration component as a pre-processing step would result in sending some words and phrases into the transliteration component which might not need to be transliterated. This would result in bad translations. Moreover the SMT system may no longer have had access to longer phrases that include names from the phrase table.

4.1.5 Transliteration Setup

To perform transliteration we use the option in the Moses decoder which outputs the untranslated words in the output of the SMT system. Given the output together with untranslated words, we use a python script which identifies the Bengali untranslated words and tags them. The python script identifies Bengali words using the range of the Bengali Unicode characters which is 0980-09FF.

After identifying Bengali words, the transliteration module proposes English transliterations for all the tagged Bengali words. The transliteration module uses the Moses SMT system to learn the mapping between characters of Bengali and English on a large parallel name list treating each letter and each word as a word and a sentence respectively.

Later the proposed English transliteration is checked against the monolingual English dictionary for matches or close string distances to some entities in the dictionary. If a match is found then untranslated Bengali words are replaced with the matched English word in the

SMT output file. If no match is found then the proposed English transliteration generated by our transliteration module is replaced with the untranslated Bengali word in the SMT output. Then the new output file with the transliteration of the untranslated words is evaluated.

4.1.6 Results of Stand-alone Transliteration module

We evaluated the performance of our transliteration module as a stand-alone system. In order to evaluate a stand-alone transliteration system, the common practice is to prepare a list of name pairs and give the source language names to the transliteration system and compare the output with the corresponding names in the target language. As mentioned in the previous chapter, we have complied 220 name pairs for the development test set and 210 name pairs for blind test set. Below in table 4.2 we provide distribution of the seen and unique names in our test sets. Seen names are those names that appeared in the training data and unique names are those that did not appear in the training data.

	Seen	Unique	Total
Dev Set	111	109	220
Test Set	115	95	210

Table 4.2: Distribution of seen and unique names

We applied our transliteration module to the test set to get a list of top-n transliteration options for each entry in the test set. The reason for providing top-n results is that having top-1 results does not always provide enough insight on how well the system performs. The results are summarized in table 4.3. The top-10 result shows the percentage of test cases whose correct transliteration could be found among first 20 outputs from the transliteration system. A similar definition goes for top-5 and top-2 results and the top-1 result shows the percentage of the test cases whose correct transliteration is the first output of the transliteration system.

	Top 1	Top 2	Top 5	Top 10
Dev Set	56 %	60 %	67 %	74 %
Test Set	54 %	59 %	65 %	71 %

Table 4.3: Performance on development and blind test set

4.2 Prepositions

A preposition is a word placed before a noun to show in what relation the noun stands with regard to another noun or verb in the same phrase. Prepositions are the words that appear at the beginning of a prepositional phrases (PP). A PP is a group of words containing a preposition, an object of the preposition, and any modifiers of the object. Prepositions can be categorized into three classes: simple prepositions, compound prepositions and phrase prepositions. Simple prepositions are for example, “at”, “by”, “for”, “from” etc. A compound preposition is made up of a set of words which starts with and acts like a preposition, for example, “in spite of”, “in favor of”, “on behalf of” etc. A phrase preposition is a simple preposition preceded by a word from another category, such as an adverb, adjective, or conjunction, for example, “instead of”, “prior to”, “because of”, “according to” etc. Sometimes a word may look like a preposition but is actually part of a phrasal verb (prepositions follow the verbs together forming phrasal verbs). These words are known as particles. E.g. “Four men held up the bank.” Here “held up” is a verb “to rob”. Therefore, “up” is not a preposition, and “bank” is not the object of a preposition. Instead, “bank” is a direct object of the verb “held up”.

Both linguistic and computational aspects of prepositions have been studied by several researchers. Jackendoff [55], Emonds [37], Rauh [104] and Pullum and Huddleston [100] have investigated the syntactic characteristics of prepositions. In the field of natural language processing, the problem of PP attachment has been a topic for research for a long time. In recent years, the problem was explored with a neural network-based approach [118] and with a syntax-based trainable approach [137]. Although past research has revealed various aspects of prepositions, there is not much semantic research of prepositions available for computational use, which requires a vigorous formalization of the semantics. A recent semantic study of prepositions for computational use, with a focus on spatial prepositions, is found in [130].

Trujillo [126] presented a pattern in the translation of locative prepositional phrases between English and Spanish. Locative prepositional phrases are those which are used to specify the physical location of an action or an object. Trujillo [126] proposed a way of exploiting this pattern in the context of a multilingual machine translation system.

Chen et al. [22] developed a rule-based and MT-oriented model refinement algorithm which tackles the structural ambiguities in prepositional phrases. They propose four different attachments according to their functionality which are noun attachment, verb attachment, sentence-level attachment and predicate-level attachment from the viewpoint of MT. Both lexical knowledge and semantic knowledge are involved in resolving attachment.

Nubel [90] describes different types of knowledge sources required for the disambiguation and translation of prepositions within a multilingual (German, English and Japanese) spoken language translation system. He argues that the traditional knowledge, such as lexical semantic information alone, does not suffice, but has to be combined with context information provided by a dialogue model in order to capture the context dependence of preposition meanings.

4.2.1 Prepositions in Bengali

In the Bengali language, there are no prepositions. English prepositions are handled in Bengali using inflections on the referenced objects and/or post-positional words after them. Inflections get attached to the referenced objects. There are numerous inflections in Bengali including -e, -y, -ye, -te, -ete, -ke, -re, -ere, -r and -er. For example, if the -te inflection is attached to the word “Canada” the inflected word becomes “Canada-te” (in Canada).

Naskar and Bandyopadhyay [89] described how to handle English prepositions during translation from English to Bengali in their paper, which we use as the basis of our preposition handling module for our Bengali to English SMT system. In order to translate from English to Bengali, they first conduct a lexical analysis of the English sentences to gather the lexical features of the morphemes using WordNet. The root words or terms including idioms and named entities together with associated grammatical information and semantic categories are then extracted during the morphological analysis phase. Then a shallow parser is used to identify the constituent phrases of the source language sentence and tag them to encode all relevant information needed to translate these phrases. Bengali synthesis rules are used to translate these phrases individually to the target Bengali language. The noun phrases and PPs are translated using example bases of syntactic transfer rules and verb phrases are translated using morphological paradigm suffix tables. Then some heuristics based on the word ordering rules of Bengali are used to arrange the target language phrases to form the target language representation of the source language sentence.

4.2.2 Challenges in Bengali SMT

Our analysis based on the output of the SMT system without preposition handling shows that although we might have some of the untranslated words in our training corpus, they are untranslated due to the inflections. For example below in table 4.4 we provide output of our baseline SMT system.

Source sentence:
কিরগিজস্তানের$_1$ পরিষদ$_2$ বাকিইয়েফকে$_3$ ভারপ্রাপ্ত$_4$ প্রধানমন্ত্রী$_5$ নির্বাচন$_6$ করেছে$_7$ ।
Kirghizstan$_1$ porishod$_2$ Bakiyefke$_3$ bharprapto$_4$ prodhan$_5$ montri$_6$ nirbachon$_7$ korecha$_8$.
Kirghizstan$_1$ of$_1$ the$_2$ federation$_2$ Bakiyef$_3$ officiating$_4$ Prime$_5$ Minister$_5$ has$_7$ elected$_6$.
SMT output:
কিরগিজস্তানের council বাকিইয়েফকে prime minister of has election.
Reference sentence:
The federation of Kirghizstan has elected Bakiyef as the officiating Prime Minister.

Table 4.4: SMT output

In the above example, although the word "Kirghizstan" and its corresponding translation was found in the phrase table, it was left untranslated because an inflectional suffix was added to it in Bengali.

4.2.3 Our Preposition Handling Module

In our prepositional module, we are using inflections in Bengali in order to handle English prepositions. We handle several Bengali inflections র -r, ের -er, য়ের -yer, কে -ke, and ে -e which can be translated into any of the English propositions "in", "of", "to" and "at" based on three rules. The rules are as follows:

$$\text{Rule 1. <Bengali word> - [(r / er / yer)]} \rightarrow \text{of <English word>} \tag{4.1}$$

$$\text{Rule 2. <Bengali word> - [(ke)]} \rightarrow \text{to <English word>} \tag{4.2}$$

$$\text{Rule 3. <Bengali word> - [(e)]} \rightarrow \text{in <English word>} \tag{4.3}$$

If any of the infections (r / er / yer) are attached to the end of the Bengali word, then the translated English word will contain the proposition "of" based on rule 4.1. For example, given Bengali text Iraqer (ending with inflection er), rule 4.1 would produce the English text "of Iraq".

Our Bengali preposition handling module works according to the following steps: First it consider all words that have any of the following inflections: (r / er / yer / ke/ e). It then removes the inflection and looks for the base word in the bilingual dictionary to check if the word exists. If the word matches, it applies the appropriate rule based on the inflection. Finally, it proposes the English preposition and the translation of the word from the bilingual dictionary lookup as a correct resolution of preposition.

We apply the preposition handling module as a post-processing step to our Bengali-English SMT output after applying the statistical translation module. Our experimental results show that only incorporating the three propositional rules has an effect on the BLEU score. The preposition handling module uses the following algorithm:

Algorithm 2 Preposition handling module

```
Given the output $T$ of SMT module
for each untranslated word $s$ in the corpus do
  if the word $s$ contain the following inflections: (r / er / yer / ke/ e) then
    Let $s'$ be word $s$ with inflection removed
    Check for $s'$ in the bilingual dictionary
    if $s'$ in dictionary then
      Apply either rule 1, 2 or 3 according to the inflection
      Propose the English preposition and the translation of the word from bilingual
      dictionary as correct resolution for word $s$
    end if
  end if
end for
```

4.2.4 Prepositional Module Setup

The preposition handling module also works as a post-processing step in our SMT system. Similar to the transliteration module, given the output from the Moses SMT system with untranslated words we identify the Bengali untranslated words and tag them. When applying the proposition handling module and transliteration module together, we process prepositions before performing transliteration since once transliterated we might lose the preposition.

After identifying untranslated Bengali words, we identify words with specific inflections. We remove the inflection from the word and check against a bilingual dictionary to match with the base form of the word. If the words match, the appropriate rule based on the inflection is applied. Then the English preposition together with English translation is replaced with the untranslated Bengali word in the SMT output. Then the new output file is evaluated. We evaluate the performance of our prepositional module as part of our SMT system in section 4.4.1.

4.3 Compound Words

Compounding of words is common in a number of languages (German, Dutch, Finnish, Greek, Bengali etc.). As words may be joined freely, it causes an increase in vocabulary size which leads to data spareness problem. This poses challenges for a number of NLP applications such as machine translation, speech recognition, text classification, information extraction, or information retrieval.

Each word consisting of more than one root-word is called a compound word. In MT, the splitting of an unknown compound word into its parts enables the translation of the compound by the translation of its parts. Compound splitting is a well-defined computational linguistics task. One way to define the goal of compound splitting is to break up foreign words, so that a one-to-one correspondence to English can be established.

4.3.1 Approaches for Handling Compound Words

While the linguistic properties of compounds are widely studied [70], there has been only limited work on empirical methods to split up compounds for machine translation between Bengali and English. Dasgupta et al. [31] gave a brief description of morphological analysis of compound words in Bengali in their paper. They use a feature unification based morphological parser which can successfully and efficiently parse compound words having inflectional suffixes and at the same time resolve ambiguities in compound words.

Dictionary lookups are the most common ways for splitting compound words. Brown [14] proposed an approach which uses a parallel corpora for compound splitting. It was limited to breaking compounds into cognates and words found in a translation lexicon which can be acquired by training a SMT system. The method improved text coverage of an example based machine translation system.

Lexicon based approaches for compound splitting were proposed by Monz and de Rijke [86] and Hedlund et al. [50] for information retrieval tasks. Compounds are broken into either the smallest or the biggest words that can be found in a given lexicon. For speech recognition, Larson et al. [71] proposed a data-driven method that combines compound splitting and word recombination. Although their approach reduces the number of out-of-vocabulary words, it does not improve speech recognition accuracy. Sjoberg and Kann [117] used a statistical approach where compound words were automatically split by a modified spell-checker.

Research on compound word splitting has been done for several languages such as German [67], and Norwegian [56]. A linguistic approach to automatic analysis of Swedish compounds was described in [33]. Splitting German compounds into their parts prior to translation has been suggested by many researchers.

Koehn and Knight [67] presented an empirical splitting algorithm that is used to improve translation from German to English. They split all words in all possible places, and considered a splitting option valid if all the parts of a splitting word exists in a monolingual corpora. They also restrict all splits to be of at least length three. They allowed the addition of -s or -es at all splitting points. If there were several valid splitting options, they chose one based on the number of splits, the geometric mean of part frequencies or based on alignment data.

Popovic et al. [99] proposed approaches of using compound splitting to improve alignment, or to joining of English compounds based on POS or alignment data prior to training. All these approaches lead to improved translation quality.

Koehn et al. [63] discussed the treatment of hyphened compounds in translation into German by splitting at hyphens and treating the hyphen as a separate token, marked by a symbol. However, there was not a significant impact on the translation results.

Stymne et al. [122] used split compounds in a factored Phrase-based SMT system with morphologically enriched POS-tags for German. A modified version of the splitting algorithm of (Koehn and Knight 2003) is used, which improved translation quality.

4.3.2 Challenges in Bengali SMT

The Bengali language has a large number of compound words. Almost all combinations of noun, pronoun and adjectives can be combined with each other. A compound word's root words can be joined by a hyphen ('-') or nothing. For example, মা-বাবা -ma-baba (mother father) in Bengali means "mother and father".

In the above example, although the words মা and বাবা and their corresponding translation "mother" and "father" might be in the phrase table of the MT system, the compound word মা-বাবা might be treated as an unknown word if its corresponding translation was not found in the phrase table. So splitting the Bengali compound word মা-বাবা into words মা and বাবা might improve translation quality of the MT system since it is more likely that their corresponding translation can be found in the phrase table.

4.3.3 Our Compound Word Handling Module

One way to define the goal of compound splitting is to break up foreign words so that a one-to-one correspondence to English can be established. Compound words are created then by joining existing words together. Our compound splitting approach can be described in the following few steps:

We first handle compound words joined by a hyphen ('-') since in Bengali most of the time the meaning of these compound words is the composition of the meaning of each root-word. For all the words with hyphens, remove the hyphen, and look for each component in a monolingual Bengali corpora. If both words are found in the corpora, we propose the two words as a replacement for the compound word. So after the first step, some compound words containing hyphens will have been replaced. We then consider all possible splits of a word into known words that exist in the monolingual corpora. We restrict known words to be of at least of length three as was also done by Koehn and Knight [67]. Then for all splitting options we compute their frequency in the monolingual corpora and compute the arithmetic mean for each proposed compound split based on their frequency. The proposed compound split with the highest arithmetic mean is selected as the correct compound split.

For example for the word রাতদিন -ratdin (night and day) , we find the following possible splitting options:

রাতদিন
রাত দিন

Now we need to decide which splitting option to pick. In the above example, we need to decide if we need to split the compound word রাতদিন into words রাত দিন or leave the compound word intact as রাতদিন . We consider a frequency based splitting metric based on the word frequency. Given the count of words in the corpus, we select the split S with the highest arithmetic mean of word frequencies of its n parts p_i:

$$argmax_S(\sum count(p_i))/n$$

So for example, for the word রাতদিন , we will have the followed, where the numbers correspond to word counts.

রাতদিন (700) → 700

রাত (900) দিন (701) → 800.5

So we pick রাত দিন as the splitting option for রাতদিন . Then we check against a bilingual dictionary between Bengali and English to see if the English counterparts of the Bengali components both exists in the bilingual dictionary. If both words exist we pick রাত দিন as the final splitting option for compound word রাতদিন .

So the compound word splitting module performs according to the algorithm 3.

Algorithm 3 Compound word splitting module

```
Given a sequence of words $T$
for each word $S$ in $T$ do
  if word $S$ contain hyphen(-) then
    remove the hyphen and split into two words $s_1$ and $s_2$
    Look for both words in a Bengali monolingual corpora
    If both $s_1$ and $s_2$ are found, replace $S$ with $s_1\_s_2$
  else
    for word $S$, consider all possible splits (into known words) containing $n$ constituents
    $s_1$ $s_n$ where $n \geq 1$ and $|s_i| \geq 3$ do
      Calculate score (arithmetic mean) for each split option
      Take the split that has the highest score
    end for
  end if
end for
```

We apply the compound nouns handling module as a pre-processing step to our Bengali-English training dataset before applying to statistical translation module. We evaluated the compound splitting module against a gold standard and also measured its impact on performance of our Bengali-English SMT system.

4.3.4 Compound Word Module Setup

The compound word splitting module also works as a pre-processing step in our SMT system. We apply our compound splitting module to all training, development and test datasets. Our module goes through each word in the dataset. First it checks if the word contains hypen (-) in the middle. A lot of Bengali words are joined together using a hypen (-) to form a compound word. If a hypen (-) is found, we then remove the hypen and split the word into two words. Then upon checking both words against our large bilingual dictionary, we propose both words as splitting options and replace the compound word with the splitting option in the dataset.

If the word does not contain any hypens then we consider all splits into known words that exist in a large monolingual corpus. The split option which has the highest arithmetical mean is selected and replaced with the compound word in the dataset. The word is left unchanged if no splitting option is available. Then the pre-processed dataset is used as the input to our SMT system.

4.3.5 Results of Stand-alone Compound module

We also evaluated the performance of our compound word splitting module separately as a stand-alone system. Similar to transliteration system in order to evaluate a stand-alone compound splitting system, the common practice is to prepare a list of compound words and their splitting options and give the compound words to the compound splitting module and compare the output with the corresponding compound words splitting options in the list. As mentioned in the previous chapter, we have complied 280 compound words and created their split option from the Bengali Prothom-Alo monolingual corpora manually for evaluation.

The stand-alone compound splitting module has an accuracy of 80 % on the test set containing the 280 compound words.

4.4 SMT Experimental Setup

Below we describe the SMT system used for conducting all the experiments for Bengali-English SMT.

The SMT system we used in our experiments is Moses [65]. In Moses, phrase translation probabilities, reordering probabilities, and language model probabilities are combined in the log-linear model to obtain the best translation e of the source sentence f. The models (or features) which are employed by the decoder are: (a) one or several phrase table(s), which model the translation direction P(f|e), (b) one or several n-gram language model(s) trained with the SRILM toolkit [121]; (c) a distortion model which assigns a penalty based on the number of source words which are skipped when generating a new target phrase, and (d) a word penalty. These different models are combined log linearly. Their weights are optimized with respect to BLEU score using the algorithm described in [95] with help of a development data set. We used a beam search decoder [62], for phrase-based SMT models to search for the best target sentence.

Language models currently in use in SMT systems can be trained using packages such as SRILM [121] or the IRSTLM [38] toolkit. IRSTLM requires about half the memory of SRILM for storing an equivalent LM during decoding. However a decoder running with SRILM permits an almost two fold improvement in translation speed over IRSTLM. Since memory requirements were not an issue for us while conducting the experiments and we preferred faster translation results, so we decided to use the SRILM tookit. We built all of our language models using the SRILM toolkit with modified Kneser-Ney discounting. In decoding we used a 4-gram language model trained on the English side of the Europarl dataset and only on our training data.

In our phrase-based SMT system, the translation models and training methods follow the standard Moses [65] setup as suggested in the Moses documentation. The script for baseline SMT is given in appendix B. The system was tuned via Minimum Error Rate Training (MERT) on the development dataset provided by the LDC corpus.

We evaluate the performance of the SMT system using the BLEU accuracy measure [98], WER (word error rate), and PER (position independent word error rate) and human evaluation.

BLEU(Bilingual evaluation understudy) is an IBM-developed metric which measures how close a candidate translation is to a reference translation by doing an n-gram comparison between both translations. BLEU is a precision measure based on n-gram counts where typically n-grams of size $n \in 1, ..., 4$ are considered. The precision is modified such that multiple references are combined into a single n-gram count vector. All hypothesis unigram, bigram, trigram and fourgram counts are collected and divided by their corresponding maximum reference counts. The clipped hypothesis counts are summed and normalised by the total number of hypothesis n-grams. The geometric mean of the modified precision scores for a hypothesis is calculated and thenmultiplied with an exponential brevity penalty factor to penalise too short translations.

WER is based on the Levenshtein distance [73]. It is calculated as the minimum number of substitutions, deletions and insertions that have to be performed in order to transform the translation hypothesis into the reference sentence. A shortcoming of the WER is that it requires a perfect word order. The word order of an acceptable sentence can be different from that of the target sentence, so that the WER measure alone could be misleading. This is the standard measure for evaluation of automatic speech recognition systems.

The word order of two target sentences can be different even though they are both correct translations. To account for this, the position-independent word error rate PER proposed by Tillmann et al. [124] compares the words in the two sentences ignoring the word order. The PER is always lower than or equal to the WER.

In the figure 4.2, we illustrate the architecture of our overall statistical machine translation system which incorporates the three Bengali specific modules- the transliteration module, preposition module and compound word module. The Bengali compound module was added as pre-processing steps and Bengali preposition module and transliteration system were added as post-processing step.

4.4.1 Impact of Transliteration Module, Propositional Module, Compound Module on the SMT System

When integrating transliteration, prepositional and compound word splitting modules, we treated compound word splitting as a preprocessing step as there were many compound words whose translations were not available in the initial translation system. Resolving the compound word resulted in known words being passed to the translation engine. We applied transliteration and prepositional modules as post-processing steps as there are a lot of untranslated Bengali out of vocabulary (OOV) words in the SMT output which can be resolved by these modules. Below in table 4.5 we provide statistics of untranslated Bengali OOV words in our development and blind test dataset.

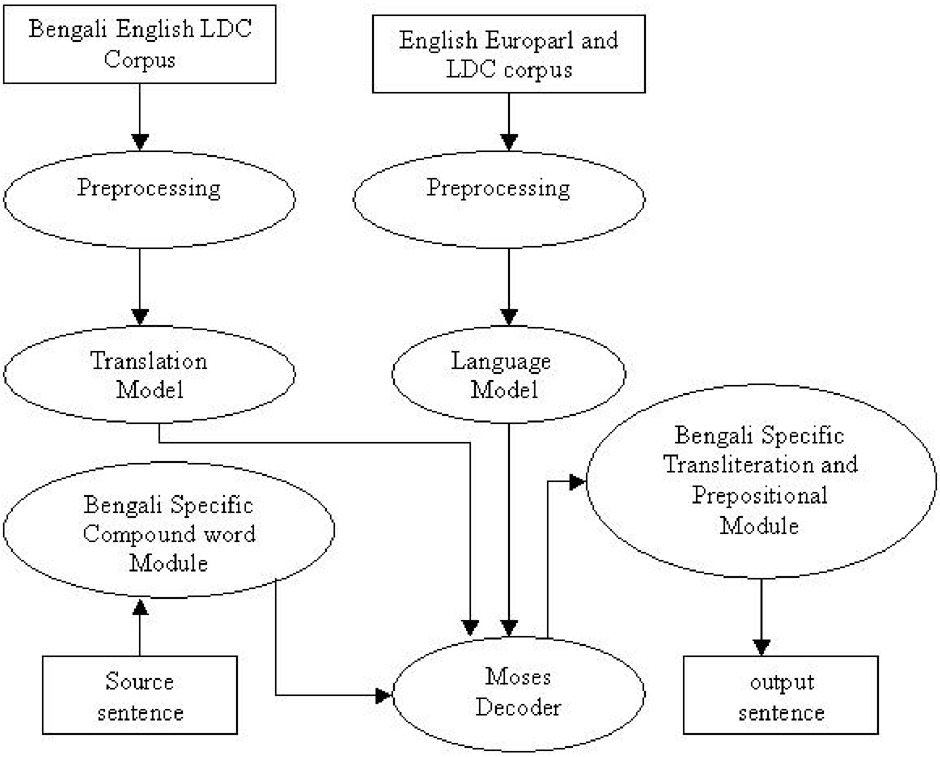

Bengali English LDC Corpus
English Europarl and LDC corpus
Preprocessing
Preprocessing
Translation Model
Language Model
Bengali Specific Transliteration and Prepositional Module
Bengali Specific Compound word Module
Source sentence
Moses Decoder
output sentence

Figure 4.2: Overall system

	Number of sentences	Total Words	OOV words	percentage of OOV words
Dev Set	600	9734	2172	22.3 %
Test Set	1000	18131	4011	22.1 %

Table 4.5: Distribution of OOV words in development and test set

Our experiments compare the translation quality of the baseline systems, baseline system with transliteration module, baseline with compound word module, baseline with preposition handling module, baseline with transliteration and prepositional module, and baseline with all three modules. We decided to use two baseline systems. The first baseline is trained on the LDC corpus and the other baseline is trained on the LDC corpus together with bilingual dictionary as mentioned in the dataset section. The results of these different approaches are shown in table 4.6 below where all the training data from LDC corpus was used to train the system and LDC test set was used to evaluate the system.

System	BLEU(%)	WER(%)	PER(%)
Baseline-1	7.2	84.7	64.6
Baseline-2 (with dictionary)	8.0	82.5	62.4
Baseline-2+Transliteration	8.6	81.1	61.5
Baseline-2+Compound words	8.3	81.9	61.9
Baseline-2+Preposition handling	8.2	82.1	62.2
Baseline-2+Preposition handling+Transliteration	8.9	80.7	61.2
Baseline-2+All	9.1	80.3	61.0

Table 4.6: Impact of transliteration, compound word and prepositional module on SMT

On the LDC corpus test set, our baseline-2 system obtains a BLEU score of 8.0 which we consider as our baseline for the rest of the experiments and incorporating the transliteration module into our translation system the BLEU score is increased from 8.0 to 8.6. Incorporating the compound module and prepositional module our system achieves BLEU score of 8.3 and 8.2 respectively. Our system obtains a BLEU score of 9.1 when all three modules are incorporated which is a 1.1 increase in BLEU score over the baseline system. When applying preposition handling module and transliteration module together as post-processing, we first apply the preposition handling module and later transliteration. The reason for that is the preposition handling module can first resolve some inflections by applying the manual rules and later the transliteration module can be applied on those entries to be transliterated if they are not available in bilingual dictionary.

4.4.2 Impact of Transliteration module, Propositional Module, Compound Module on New Extended Testset

We evaluated the translation quality of our SMT system on the new extended test set. In table 4.7, we provide the results when two reference test sets are used and in table 4.8 when three reference test sets are used. We achieve on average between 2.3 and 4.2 increase in BLEU score with two and three reference test sets respectively compared to the single reference test set provided by the LDC corpus. We also achieve a significant decrease in the WER and PER score with the new extended test set.

System	BLEU	WER	PER
Baseline(with dictionary)	10.3	77.2	56.3
Baseline+Transliteration	10.9	76.2	55.5
Baseline+Compound words	10.6	76.9	56.1
Baseline+Preposition handling	10.5	76.5	55.8
Baseline+All	11.1	76.1	55.2

Table 4.7: Impact of transliteration, compound word and prepositional module on SMT using two test reference

System	BLEU	WER	PER
Baseline(with dictionary)	12.2	74.8	54.1
Baseline+Transliteration	12.8	73.3	52.8
Baseline+Compound words	12.4	74.4	53.7
Baseline+Preposition handling	12.2	74.7	54.0
Baseline+All	13.0	73.1	52.7

Table 4.8: Impact of transliteration, compound word and prepositional module on SMT using three test reference

4.5 Word Reordering

The difference in word order between languages is one of the main difficulties a MT system has to deal with. These can range from small differences in word order as is the case between most European languages to a completely different sentence structure such is the case of translation from Bengali into English.

Word reordering strategies within SMT allow for improvement in translation accuracy when source and target languages have significantly different word order. The reordering of words in machine translation still remains one of the challenging problems. We apply word reordering to the Bengali-English machine translation task. The MT task can be divided into two sub tasks: one is predicting the translation of words and the second is deciding the order of words in the target language. For some language pairs such as Bengali-English, Japanese-English, the reordering problem is hard since the target language word order differs significantly from the source word order. Reordering often depends on higher level linguistic information, which is absent in phrase-based SMT.

Bengali grammar generally follows the SOV structure and English follows the SVO structure. Since Bengali (SOV) and English (SVO) have different sentence structure, phrase reordering is indeed important for this language pair to achieve quality translation. Different languages differ in their syntactic structure. These differences in word order can be

1) local word reordering and

2) global/long-range word reordering

Local word reordering: Local reordering includes the swapping of adjective and noun in language pairs like Spanish and English. Below in table 4.9 are the examples of some possible local reordering in Bengali.

বোমা হামলার আগে ⇒ আগে বোমা হামলার
bomb attack before ⇒ before bomb attack

Table 4.9: Local word reordering

Global word reordering: Global reordering can involve long range verb movement, since the position of the Bengali verb is different from the one in the English sentence in many cases. The verb in the Bengali sentence must be moved from the end of the sentence to the beginning just after the subject in the English sentence. Bengali-English SMT can benefit from both local and global reordering on the Bengali side. Below in table 4.10 are the examples of some possible global reordering in Bengali.

কিরগিস্তানের পরিষদ বাকিইয়েফকে ভারপ্রাপ্ত প্রধানমন্ত্রী নির্বাচন করেছে ।
The federation of Kirghizstab Bakiyef as the officiating Prime Minster has elected.
কিরগিস্তানের পরিষদ নির্বাচন করেছে বাকিইয়েফকে ভারপ্রাপ্ত প্রধানমন্ত্রী।
The federation of Kirghizstab has elected Bakiyef as the officiating Prime Minster.

Table 4.10: Global word reordering

4.5.1 Word Reordering Approaches

A number of researchers ([12], [9], [93], [133], [26]) have described approaches that preprocess the source language input in SMT systems. We are not, however, aware of work on this topic for translation from Bengali to English. Brown et al. [12] describe an analysis component for French which moves phrases around (in addition to other transformations) so the source and target sentences are closer to each other in word order. Berger et al. [9] describe an approach again for French that reorders phrases of the form NOUN1 de NOUN2, while Xia and McCord [133] describe an approach for French, where reordering rules that operate on context-free rule productions are acquired automatically. Niessen and Ney [93] describe an approach for translation from German to English that combines verbs with associated particles, and also reorders questions. Collins et al. [26] also describe an approach for German, concentrating on reordering German clauses, which have quite different word order from clauses in English.

Unlike local reordering models that emphasize the reordering of adjacent phrase pairs [125], our proposed model will focus explicitly on modelling the long range reordering. Since phrase-based systems have relatively limited potential to model word-order differences between different languages in our approach the reordering stage attempts to modify the source language (e.g. Bengali) in such a way that its word order is very similar to that seen in the target language (e.g., English) based on the automatically learning reordering from POS tagged source language.

Fei Xia et al. [133] describe an approach to automatically acquire reordering rules in translation from French to English. The reordering rules operate at the level of context-free rules in the parse tree. Similar approaches ([19], [91]) also propose the use of rules automatically extracted from word aligned training data. The results in their studies show that translation performance is significantly improved in BLEU score over baseline systems.

Collins et al. [26] applied a sequence of hand crafted rules to reorder the German sentences in six reordering steps: verb initial, verb 2nd, move subject, particles, infinitives, and negation. This approach successfully shows that adding syntactic knowledge can represent a statistically significant improvement from 1 to 2 BLEU points over baseline systems.

Xu et al. [134] presented a novel precedence reordering approach based on a dependency parser to SMT systems. They claim that their approach can efficiently incorporate linguistic knowledge into SMT systems without increasing the complexity of decoding. For a set of five SOV order languages, they applied this approach to systems translating English to Korean, Japanese, Hindi, Urdu and Turkish. They proposed precedence reordering rules based on a dependency parse tree. All rules were based on English and Korean examples. These rules were extracted manually by a bilingual speaker after looking at some text book examples in

English and Korean, and the dependency parse trees of the English examples. A precedence reordering rule is a mapping from T to a set of tuples (L, W, O), where T is the part-of-speech (POS) tag of the head in a dependency parse tree node, L is a dependency label for a child node, W is a weight indicating the order of that child node and O is the type of order (either NORMAL or REVERSE). Given a set of rules, they are applied in a dependency tree recursively starting from the root node. If the POS tag of a node matches the left-hand-side of a rule, the rule is applied and the order of the sentence is changed. They describe three different kind of precedence rules to reorder English to SOV language order which are verb precedence rules, adjective precedence rules and noun and preposition precedence rules. For all 5 languages, they achieve statistically significant improvements in BLEU scores over a state-of-the-art phrase-based baseline system. However, for Korean and Japanese, their precedence reordering rules achieve better absolute BLEU score improvements than for Hindi, Urdu and Turkish.

Many publications deal with the word reordering problem, but only a few make use of linguistic knowledge about the sentence structure. Nießen and Ney [92] proposed an approach for using morpho-syntactic information for word reordering in SMT for the German–English pair. They proposed two reordering transformations which are: prepending German verb prefixes to the main verb and inversion of interrogative sentences using syntactic information.

In the last few years several publications addressed the problem of local reordering for the Spanish–English language pair. In [72], reordering rules are acquired from a word aligned parallel corpus using POS tags of the source part and then applied as a preprocessing step. A similar method for extracting local reordering patterns for both translation directions is explored in ([83], [29]). The obtained patterns are then used for the creation of word graphs which contain all possible paths. A similar approach for the Chinese–English language pair is presented in [139], but shallow parsing chunks for phrase reordering are used instead of POS tags for word reordering.

Extracting rules from word alignments and source language POS tags is also presented in [106] for the Spanish–English and German–English language pair. These rules are then used for the creation of word graphs, but the graphs are extended with the word or POS tag context in which a reordering pattern is seen in the training data. The reordering rules are extracted from word alignments along with automatically learnt word classes in [27] for the Spanish–English language pair.

4.5.2 Bengali Word Reordering

Although statistical word alignments work rather well at capturing differences in word order and a number of strategies for non-monotonic search have been developed, differences in word order between the source and the target language are still one of the main causes of translation errors. We investigate possibilities for improving the translation quality by rule-based reordering of the source sentence using only the POS information of the source language.

The Bengali language has many particularities which differences it from English language. Those differences makes the translation between English and Bengali an interesting challenge which involves both morphological and syntactic features. As noted earlier, Bengali is also low resourced which makes the development of a SMT system even more difficult.

A word is "reordered" when it and its translation occupy different positions within the corresponding sentence. In this section we consider reordering between Bengali and English based on POS-based word reordering. This approach requires only the POS information of the source language. A parse tree or some other type of detailed information about syntax is not necessary.

We experimented with reordering the Bengali training data into an English-like word order before running Moses training. When translating an unseen Bengali sentence to English, we first preorder it into this English-like word order, then translate the preordered Bengali sentence with the specially-trained Moses setup. With this approach, the burden of reordering phrases is pushed to a syntactic preprocessing step, and the Moses translator itself can perform a largely monotonic (no reordering) translation, at which it excels. The challenge is to build methods that reorder a Bengali sentence into a pseudo-Bengali sentence that has the same words but in English-like word order.

In this thesis we describe two such approaches. In one approach reordering is done on automatically learnt rules from an aligned training corpus and in the second approach reordering is done on a predefined set of rules identified by linguists. We also experimented with lexicalized reordering implemented in Moses. Below we describe lexicalized reordering, automatically extracted rules reordering and manually extracted rules reordering approaches.

4.5.3 Lexicalized Reordering

First we experimented with lexicalized reordering[6] implemented in Moses. Although this approach does not involve any preprocessing of the source side, it does add new features to the log-linear framework, in order to determine the order of the target phrases during decoding.

The default phrase-based statistical machine translation model is only conditioned on movement distance. However, some phrases are reordered more frequently than others. A French adjective like "extérieur" typically gets switched with the preceding noun, when translated into English. Therefore a lexicalized reordering model that conditions reordering on the actual phrases is beneficial. However there is problem of data sparseness which makes it hard to reliably estimate probability distributions when a particular phrase pair only occurs a few times in the training data. So the lexicalized reordering model implemented in Moses only considers three reordering types: (m) monotone order, (s) switch with previous phrase, or (d) discontinuous. Figure[7] 4.3 below illustrates these three different types of orientations of a phrase.

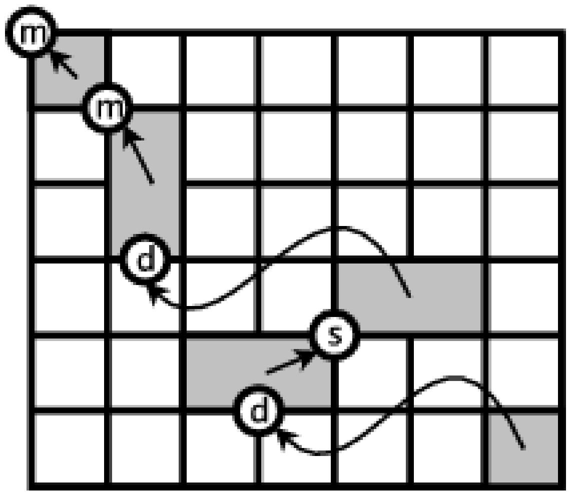


Figure 4.3: Lexicalized reordering

More formally a reordering model p_o is introduced that predicts an orientation type $\{m, s, d\}$ given the phrase pair currently used in translation:

[6] http://www.statmt.org/moses/?n=Moses.AdvancedFeatures

[7] Figure taken from http://www.statmt.org/moses/

$orientation \epsilon \{m, s, d\}$

$p_o(orientation|f, e)$

Such probably distributions can be learnt from the word alignment which is the basis of the phrase table. When each phrase pair is extracted, its orientation type can also be extracted in that specific occurrence.

While extracting phrase pairs from the training corpora its orientation type is also extracted and the probability distribution is estimated in order to be added to the log-linear framework. Finally during decoding, automatically inferred reordering models are used to score each hypothesis according the orientation of the used phrases.

For the word alignment matrix, for each extracted phrase pair its corresponding orientation type can be detected. The orientation type can be detected, if we check for a word alignment point to the top left or to the top right of the extracted phrase pair. An alignment point to the top left signifies that the preceding English word is aligned to the preceding Foreign word. An alignment point to the top right indicates that the preceding English word is aligned to the following French word as illustrated in figure[8] 4.4.

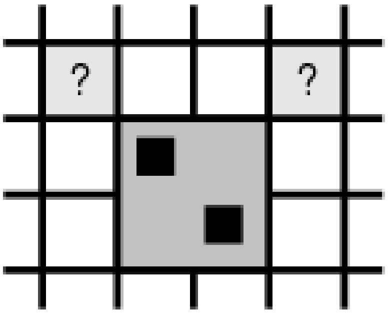


Figure 4.4: Lexicalized reordering training

The orientation type is defined as follows:

- monotone: a word alignment point to the top left exists
- swap: a word alignment point to the top right exists
- discontinuous: no alignment points to the top left or top right exists

We count how often each extracted phrase pair is found with each of the three orientation types. The probability distribution p_o is then estimated based on these counts using the maximum likelihood principle:

[8] Figure taken from http://www.statmt.org/moses/

$$p_o(orientation|f,e) = count(orientation,e,f)/\Sigma_o count(o,e,f) \tag{4.4}$$

Given the sparse statistics of the orientation types, we may want to smooth the counts with the unconditioned maximum-likelihood probability distribution with some factor Σ:

$$p_o(orientation) = \Sigma_f \Sigma_e count(orientation,e,f)/\Sigma_o \Sigma_f \Sigma_e count(o,e,f) \tag{4.5}$$

$$p_o(orientation|f,e) = (\sigma p(orientation) + count(orientation,e,f))/(\sigma + \Sigma_o count(o,e,f)) \tag{4.6}$$

There are a number of variations of this lexicalized reordering model based on orientation types:

• bidirectional: Certain phrases may not only flag, if they themselves are moved out of order, but also if subsequent phrases are reordered. A lexicalized reordering model for this decision could be learned in addition, using the same method.

• f and e: Out of sparse data concerns, we may want to condition the probability distribution only on the foreign phrase (f) or the English phrase (e).

• monotonicity: To further reduce the complexity of the model, we might merge the orientation types swap and discontinuous, leaving a binary decision about the phrase order.

These variations have shown to be occasionally beneficial for certain training corpus sizes and language pairs. Moses allows the arbitrary combination of these decisions to define the reordering model type (e.g. bidrectional-monotonicity-f).

4.5.4 Automatic Reordering

The automatic reordering considers the reordering preprocessing as the translation of the source sentences into a reordered source language, which allows a better translation into the target language.

In our approach we extract syntactic reordering rules from a parallel training corpus with a tagged source side similar to [29]. These syntactic rules are used for reordering before the translation task.

The syntactic reordering rules are learned from an aligned corpus, containing word-to-word alignments, for which the POS information of the source sentences is available. Rules are extracted based on identifying all the crossings produced in the word-to-word alignments. After a crossing has been detected, its source POS tags and alignments are used to generate reordering rules. Then generated reordering rules are applied to all source sentences.

Learning reordering Rules

In this framework, the first step is to extract reordering rules. Therefore, an aligned parallel corpus and the POS tags of the source side are needed. For every sequence of source words where the target words are in a different order, a rule is extracted that describes how the source side has to be reordered to match the target side. A rule may for example look like the following:

$$\text{NAME N ADJ V} \rightarrow \text{V NAME N ADJ} \tag{4.7}$$

The framework can handle rules that only depend on POS tags. We will refer to these rules as our reordering rules.

The rules that are later applied to the source sentences are learned via an aligned corpus for which the POS information of the source is available. Given a sentence pair with source word f_1^J and target words e_1^I and the alignment a_1^J a reordering rule is extracted whenever the alignment contains a crossing, i.e. whenever there is i and j with $i < j$ and $a_i > a_j$.

Let us consider the following example containing two cross alignments to demonstrate how reordering rules are extracted:

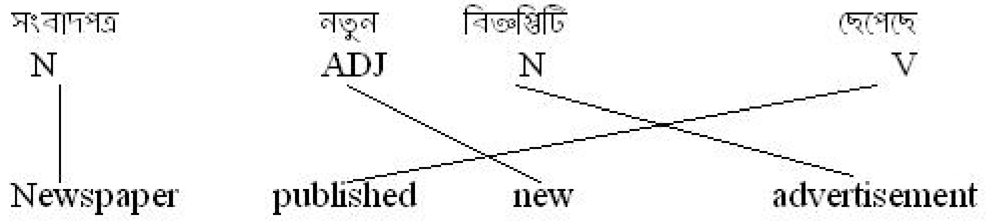


Figure 4.5: Reordering example

Based on the POS tags, the following two rules are extracted, one for each cross alignment.

Rule 1: N V ⇒ V N

Rule 2: ADJ N V ⇒ V ADJ N

The next step is to calculate the frequencies of rules. The frequencies are calculated from the number of times any particular rule is observed in the source side of the training data. Using the frequencies of the reordering rules we filter out rules that are observed less than 10 times in order to obtain only the most frequent rules. We choose the rules that were observed more than 10 times based on the experiments we conducted on different settings of the frequency of rules. The table 4.5.4 shows some of the most frequent reordering rules extracted from the training data and their frequency.

Source sequence	Rule	Frequency
N ADJ	ADJ N	256
N ADJ V	V N ADJ	85
NAME N ADJ V	V NAME N ADJ	70
ADJ N PREP	PREP ADJ N	56

Table 4.11: Reordering rules

Applying the reordering Rules

We apply all the reordering rules and reorder the source sentences of the training data in order to obtain a monotone alignment. For each rule automatically extracted from the training data, we check if any POS sequence of the source side of the training data matches the rules. If any sequence matches the rules, we apply that rule to the source side of the training data. After applying all the reordering rules we achieve a reordered source side of the training data. We train a state-of-the-art SMT system from the reordering source sentences. Similarly we also reorder the development set and the test set data. Figure 4.6 describes the overall reordering approach.

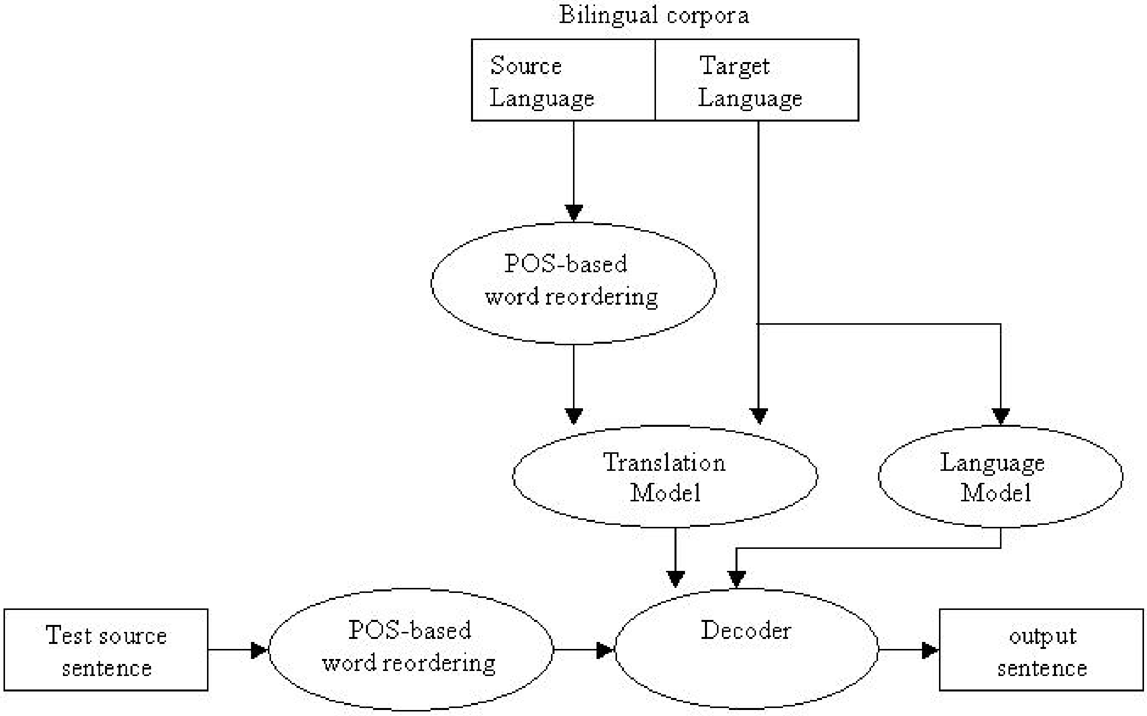


Figure 4.6: Reordering approach

4.5.5 Reordering with Manually Created Rules

We examined the Bengali source side sentences of the training data tagged with POS and manually detected several reordering rules. We apply these manually extracted reordering rules to reorder the source side of the training data and train a state-of-art SMT system on the reordered source side. Below we describe these manually extracted reordering rules.

Negative sentences

In Bengali the negative particle "না " follows the finite verb to form the negative of all tenses except the present perfect and the past perfect. For example,

$$\text{আমি যাব না} \rightarrow \text{I shall go not} \rightarrow \text{I shall not go} \tag{4.8}$$

To form the negative of the perfect tenses the particle "নাই " is added to the present tense of the verb.

$$\text{আমি যাই নাই} \rightarrow \text{I go didn't} \rightarrow \text{I didn't go} \tag{4.9}$$

We propose some reordering rules that move the negative particle in front of verb.

$$\begin{aligned} &\text{V না} \rightarrow \text{না V} \\ &\text{V নাই} \rightarrow \text{নাই V} \end{aligned} \tag{4.10}$$

Questions

A sentence may be a question in Bengali by the appearance of the unstressed interrogative particle “কি ” before or after the verb. For example,

$$\text{তুমি কি করেছ} \rightarrow \text{you what have done} \rightarrow \text{what have you done} \tag{4.11}$$

We propose some reordering rules that move interrogative particles such as “কি”, “কেন”, “কেমনে”, “কোথায় ” in front of the sentence.

$$\begin{aligned} &* \text{ কি V} \rightarrow \text{কি } * \text{ V} \\ &* \text{ V কি} \rightarrow \text{কি } * \text{ V} \\ &* \text{ কেন V} \rightarrow \text{কেন } * \text{ V} \\ &* \text{ V কেন} \rightarrow \text{কেন } * \text{ V} \\ &* \text{ কেমনে V} \rightarrow \text{কেমনে } * \text{ V} \\ &* \text{ V কেমনে} \rightarrow \text{কেমনে } * \text{ V} \\ &* \text{ কোথায় V} \rightarrow \text{কোথায় } * \text{ V} \\ &* \text{ V কোথায়} \rightarrow \text{কোথায় } * \text{ V} \end{aligned} \tag{4.12}$$

Prepositions in Bengali

In Bengali, there is no concept of preposition as we saw earlier. English prepositions are handled in Bengali using inflections to the reference objects or post-positional words after them.The post-positional words are independent words. They have meanings of their own and are used independently like other words. A post-positional word is positioned after an inflected noun (the reference object). Some examples of the post positional words in (colloquial) Bengali are: (দিয়ে [by]), (থেকে [from]), (জন্য [for]), (কাছে [near]), (আগে [before]), (নিচে [under]), (উপরে [on]) etc.

দুর্ঘটনার আগে → the accident before → before the accident
পাহাড়ের উপর → the hill on → the hill on (4.13)
সমুদ্রের নিচে → Under the sea → the sea under

We propose the below reordering rule that move the post-positional word in front of the nouns in the Bengali sentences to make it closer to English sentences.

$$\text{N PREP} \rightarrow \text{PREP N} \tag{4.14}$$

Verbs

Bengali, being a SOV language, has its verbs at the end of the sentence, unlike English which is subject-verb-object. For example,

জরুরী বিজ্ঞপ্তি প্রকাশ → important statement issued → issued important statement (4.15)

Based on the example we propose the one 'automatic' reordering rule that moves the verb to the front of the object.

$$\text{ADJ N V} \rightarrow \text{V ADJ N} \tag{4.16}$$

We then propose the following selection of manual rules:

$$\begin{gathered}
\text{N ADJ V} \rightarrow \text{V N ADJ} \\
\text{ADV N V} \rightarrow \text{V ADV N} \\
\text{N V} \rightarrow \text{V N} \\
\text{ADJ V} \rightarrow \text{V ADJ} \\
\text{ADV V} \rightarrow \text{V ADV} \\
\text{ADV ADJ V} \rightarrow \text{V ADV ADJ} \\
\text{NAME V} \rightarrow \text{V NAME}
\end{gathered} \tag{4.17}$$

The main motivation of the above manual verb reordering rules is to move verbs from the end of Bengali sentences to make them closer to English sentences.

The main difference between the manual and automatic reordering rules is that the manual rules are linguistically motivated and created by a Bengali language expert, whereas automatic reordering rules are based on cross alignment and might not be always correct.

4.5.6 Reordering Module Setup

The word reordering module also works as a pre-processing step in our SMT system. We apply our word reordering approach to all training, development and test datasets. We learn POS-based reordering rules via an aligned training corpora for which the POS information of the source is available. Then learnt reordering rules are filtered out using relative frequencies of the reordering rules. Any rule that is observed less than 10 times is ignored.

These POS-based reordering rules are then used to reorder training, development and test dataset. For each sentence in the dataset, if any rule matches the POS tag of any sentence, then based on the POS-based rule that sentence is reordered. Finally the reordered training, development and test set is used as the input to our SMT system.

We also use our manually created POS-based reordering rules to reorder the dataset. And lexicalized reordering (e.g. bidirectional-monotonicity-f) implemented in Moses is also used as a reordering option.

4.5.7 Results on Reordering Approaches

We evaluated the translation quality of our SMT system on the three reordering approaches: lexicalized reordering, automatic reordering approach and manual reordering approach. Table 4.12 summarizes the results of our manual reordering and automatic reordering rules.

	Automatic rules	Manual rules
num. of extracted rules	6350	20
num. of rules after filtering	3120	20
average length of a rule	3.4	2.8

Table 4.12: Reordering rules statistics

In tables 4.13 we present the result of the reordering approaches evaluated on a single reference test set. Reordering the sentences using automatic reordering rules contributed the most improvement towards Bengali-English SMT system which achieves an improvement of 1.4 BLEU score over the baseline. The manual reordering approach achieves a higher BLEU score over lexicalized reordering and the baseline approach. For lexicalized reordering, we experimented with different possible configurations of variations of the lexicalized reordering model such as bidirectional-monotonicity-f, monotonicity-f, msd-bidirectional-fe etc. Our experimental results indicate that msd-bidirectional-fe option has the most impact on the translation quality which we used as our lexicalized reordering option.

method	BLEU	WER	PER
Baseline(no reordering)	8.0	82.5	62.4
Lexicalized reordering	8.2	81.3	61.2
Manual reordering approach	8.4	81.8	61.9
Automatic reordering approach	9.4	80.5	60.1

Table 4.13: Impact of reordering approaches on SMT

In table 4.14 we report the BLEU score of the reordering approaches evaluated on our newly extended reference test set which contains two and three references for the test set.

method	BLEU(2 ref)	BLEU(3 ref)
Baseline	10.3	12.2
Lexicalized reordering	10.4	12.4
Manual reordering	10.6	12.5
Automatic reordering	11.3	13.8

Table 4.14: Impact of reordering approaches on SMT using two and three test references

In table 4.15 we provided the results in BLEU score of our overall model which include all the Bengali specific modules and automatic reordering module together and the baseline model evaluated on single, two and three reference test set.

method	BLEU(1 ref)	BLEU(2 ref)	BLEU(3 ref)
Baseline	8.0	10.3	12.2
Our System	10.2	12.4	14.9

Table 4.15: Impact of all approaches on SMT

4.6 Bengali Part of Speech Tagger

The framework used for creating the Bengali part of speech (POS) tagger was centered around the MALLET toolkit [84]. MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text. We used the POS tagged data provided by the Research on English and Foreign Language Exploitation- Less Commonly Taught Languages (REFLEX-LCTL) program [116]. There are 17 unique POS tags and around 27970 POS tagged words available for training in the dataset. Below we list the 17 POS tags used in the data.

POS Tags	Description	POS Tags	Description
N	noun	V	verb
PUNC	punctuation	ADJ	adjective
NAME	name/proper noun	PRON	pronoun
AUX	auxilary	CONJ	conjugate
ADV	adverb	NLOC	noun location
NUM	numeral	PREP	preposition
QF	quantifier	NEG	negation
PART	particle	QW	question word
INT	intensifier		

Table 4.16: POS tags

We used the Maximum Entropy Markov Model(MEMM) algorithm in the MALLET toolkit to train the Bengali POS tagger model. We evaluated the tagger on the test data which contains around 5582 tagged words provided by the REFLEX-LCTL program. The tagger has an accuracy of 88 %.

4.7 Morphological Analyzer

Morphology is the field of the linguistics that studies the internal structure of the words. Morphological analysis is an essential step in any NLP application. Morphological analysis takes a word as input and identifies their stems and affixes. It provides information about a word's semantics and the syntactic role it plays in a sentence. It is essential for Bengali as it has a rich system of inflectional morphology as do other Indo-Aryan family languages.

A morphological analyzer is a program for analyzing the morphology of a given word. Morphological analyzers usually use lexicon/thesaurus, keep/stop lists, and indexing engines for analyzing the words. A morphological analyzer detects morphemes in a piece of text, and is sometimes applied as a pre-processing step in NLP applications. One of the common preprocessing steps is stemming. Stemming is a process through which suffixes are removed and words are converted to their stems. For example, the word “running” might be stemmed to “run”. Stemming rules for English language are simple. Many other languages like Bengali have more complex morphosyntactic characteristics such as different suffixes or prefixes that can be used with a single word depending on the tense, gender, number, case, etc. More complex rules are needed for their stemming for these languages than for English.

4.7.1 Bengali Morphological Analyzer

We developed a morphological analyzer for Bengali that contains hand-written rules about the Bengali language morphosyntactic structure and uses that knowledge to detect morphemes.

The rule-based morphological analyzer has lists of noun stems, verb stems, noun suffixes, verb suffixes, indeclinable words collected from the REFLEX-LCTL. In the table below we provide examples of some of the noun and verb suffixes.

Noun suffixes	Description
কে	Accusative Noun Suffix
তে	Locative Noun Suffix
টা	Definite Singular Noun Suffix
টি	Diminutive Singular Noun Suffix
গুলো	Definite Plural Noun Suffix
গুলি	Diminutive Plural Noun Suffix
তে	Locative Singular Noun Suffix
য়ে	Locative Singular Noun Suffix
রা	Definite Plural Nominative Animate Noun Suffix
দের	Definite Plural Genitive Animate Noun Suffix

Table 4.17: Noun suffixes

Verb suffixes	Description
তে	Infinitive Verbal Suffix
ি	1st Person Present Verbal Suffix
ছি	1st Person Present Continuous Verbal Suffix
লে	2nd Person Ordinary Simple Past Verbal Suffix
লেন	2nd/3rd Person Honorific Simple Past Verbal Suffix
তাম	1st Person Imperfect Conditional Verbal Suffix
ছিল	3rd Person Ordinary Past Continuous Verbal Suffix
চ্ছি	1st Person Present Continuous Verbal Suffix
চ্ছিল	3rd Person Honorific Past Continuous Verbal Suffix
নোর	Genitive Verbal Noun Verbal Suffix

Table 4.18: Verb suffixes

The morphological analyzer expects input to be POS-tagged prior to morphological analysis which is achieved using our Bengali POS tagger. The POS tagged input helps to prevent applying verb suffixation to nouns.

The morphological analyzer first identifies the POS tag of a given word. Based on the POS tag, the corresponding suffix and stem lists are checked. If an observed suffix string has previously been observed with the same POS tag then a morphological split is made. The morphological analyzer does not attempt to split words found in the list of indeclinables. Indeclinable are those words that are not derived from a root and affix combination.

We evaluated the performance of the morphological analyzer on a stand-alone test set of 3518 Bengali words. Out of the 3518 words, the morphological analyzer is able to analyze 3152 words correctly. The coverage of the morphological analyzer is 89 %.

The rich inflectional morphology of the Bengali language poses problems, especially for translation with scarce resources. The full forms of the Bengali words usually contain information which is not always relevant for translation into English. Therefore we convert all Bengali words into their base forms using the Bengali morphological analyzer. The results in table 4.19 demonstrate the impact of the Bengali morphological analyzer on our Bengali-English SMT system on a single reference test set.

System	BLEU	WER	PER
Baseline	8.0	82.5	62.4
base word from	8.4	81.6	60.5

Table 4.19: Impact of morphological analyzer

4.8 Factored Model

We used factored translation models [64] for SMT, which incorporate multiple levels of information as first discussed in section 2.1.3. These layers of information, or factors, are integrated into both the training data and the models. The parallel corpora used to train factored translation models are tagged with factors such as parts of speech and lemmas. Instead of modeling translation between fully-inflected words in the source and target, the factor model incorporates more general mappings between factors in the source and target and also between factors within the target language.

Factored translation models can better handle morphology, which is one of the short-comings of the traditional SMT systems. In a traditional SMT model each word form is treated as a token in itself. This means that the translation model treats for example, the word “school” as completely independent of the word “schools”. Any instance of “school” in the training data does not add any knowledge to the translation of the word “schools”. While the translation of “school” may be known to the model, the word “schools” may be unknown and the system will not be able to translate it. This problem does not show up that often in English because of very limited morphological variation. However, it does constitute a significant problem for morphologically rich languages such as Arabic, Bengali, German and Czech.

Therefore when translating between morphologically rich languages or translating from morphologically rich languages to English, a factored translation model can be beneficial. In a factored model, for example lemma and morphological information would be translated separately and then combined on the target side to generate the ultimate output surface words. Factored models makes more efficient use of the translation lexicon. The translation of the factored representation of source words into the factored representation of target words is broken up into a sequence of mapping steps that either translate input factors into output factors known as the translation step, or generate additional target factors from existing target factors known as the generation step.

We applied a factored model to our Bengali-English SMT system by using the Moses SMT system which can handle factored models. On the Bengali side the factors we used were surface form, lemma and part of speech tags. As English factors we used surface form and part of speech tags. We applied different combinations of factors for the translation model such as source lemma to target surface form, source surface form, lemma to target surface form. Table 4.20 below illustrates the results of applying different combination of factors to the Bengali-English SMT. In factored model 1 we use Bengali factors surface form, lemma and English factor surface form. In factored model 2 we use Bengali factors surface form, lemma, part of speech tags and English factor surface form, part of speech tags and finally in factored model 3 we use Bengali factors surface form, lemma, part of speech tags and English factor surface form which contributes most to the BLEU score.

System	BLEU	WER	PER
Baseline-non factor	8.0	82.5	62.4
Factored model 1	7.9	82.2	61.9
Factored model 2	8.2	82.1	61.8
Factored model 3	8.5	81.5	61.1

Table 4.20: Impact of factored model on SMT

4.9 Manual Evaluation

In the results we have presented so far, the focus has been on automatic evaluation schemes like BLEU, WER and PER. We now propose a new manual evaluation approach for evaluating MT output which does not require source language knowledge, and as a consequence requires less time of a human evaluator as compared to other manual evaluation techniques. Since automatic evaluation like BLEU is not always sufficient to reflect a genuine improvement in translation quality, the manual evaluation approach can be used as complement of automatic evaluation methods.

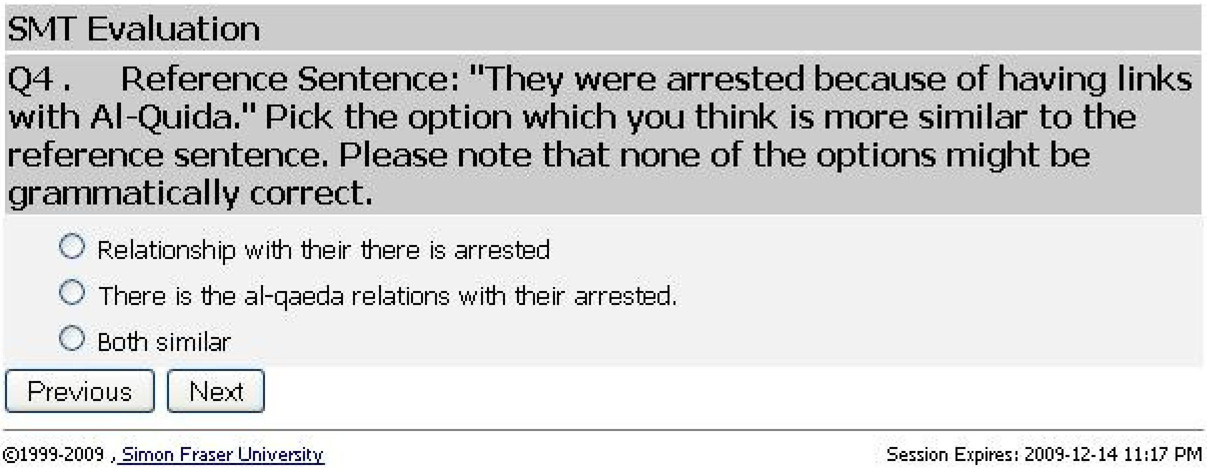
In addition to automatic MT evaluation using BLEU, PER and WER we conducted a manual MT quality evaluation. Since manual evaluation is time consuming, we are only comparing our overall SMT system containing transliteration, prepositional and compound word module and word reordering module with the baseline system. In table 4.21 we provided the results of the BLEU, PER and WER score for our overall model which include all the Bengali specific modules and automatic reordering module together and the baseline model in order for comparison with manual evaluation.

System	BLEU	WER	PER
Baseline	8.0	82.5	62.4
Our System	10.2	80.1	59.1

Table 4.21: Result of SMT

We randomly selected 20 English sentences from the test set output of our overall system, and the baseline system. We then created a survey with 20 questions where each question is the reference sentence and each answer contains three options – baseline output, our overall system output or a statement saying both outputs are similar. Then we ask humans to choose the option which is most appropriate for the reference sentence.

An example question from the survey is as follows:



SMT Evaluation

Q4 . Reference Sentence: "They were arrested because of having links with Al-Quida." Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

○ Relationship with their there is arrested

○ There is the al-qaeda relations with their arrested.

○ Both similar

©1999-2009 , Simon Fraser University

Session Expires: 2009-12-14 11:17 PM

Figure 4.7: Sample of survey

We sent the survey to 22 participants and compiled their responses for the 20 questions of the survey. In table 4.22 we present the survey results displaying users preference of the systems.

System	Preference
Baseline	22.3 %
Our System	52.3 %
Both Similar	25.4 %

Table 4.22: Survey results user preference

Based on the 20 questions of the survey we performed further analysis of the results and divided the question into two categories. One was for shorter questions with the sentence length of 10 words or less, and the other for longer questions with the sentence length of 20 words of more. Below are the results describing which category of sentences where handled better by which system according to the survey results.

System	Short sentences	Long sentences
Baseline	38.63	13.22
Our System	25.00	63.63
Both Similar	36.36	23.14

Table 4.23: Survey results of sentence categories

Based on the survey results, the baseline system was slightly better for shorter sentences, however our system outperforms the baseline system in handling longer sentences.

4.10 Summary

In this chapter, we described the transliteration, compound word, preposition and word reordering module that we incorporated into the Bengali SMT system. We performed detailed experimental evaluation of these modules using automatic and manual evaluation techniques. We found that incorporation of the modules resulted in an improvement to translation quality as reflected in BLEU, PER and WER scores. Furthermore a new manual evaluation technique showed that our system output was significantly better. We also investigated the use of factored models in our system.

Chapter 5

Machine Learning Approaches for Bengali SMT

In this chapter we describe two machine learning approaches for Benagli SMT. We will first examine semi-supervised learning, and then see how it can be used in Bengali to English MT. We will then examine active-learning, and its application to the same MT task.

5.1 Semi-supervised Learning

Semi-supervised learning refers to the use of both labeled and unlabeled data for training. Semi-supervised learning techniques can be applied to SMT when a large amount of bilingual parallel data is not available for language pairs. Sarkar et al. [111] explore the use of semi-supervised model adaptation methods for the effective use of monolingual data from the source language in order to improve translation accuracy.

Self-training is a commonly used technique for semi-supervised learning. In self-training a classifier is first trained with a small amount of labeled data. The classifier is then used to classify the unlabeled data. Typically the most confident unlabeled points, together with their predicted labels, are added to the training set. The classifier is retrained and the procedure repeated. Note the classifier uses its own predictions to teach itself. The procedure is also called self-teaching or bootstrapping.

5.1.1 Semi-supervised Learning approaches

Recently the availability of monolingual corpora in the source language has been shown to improve the translation quality in SMT [111]. It has been also shown that adding large amounts of target language text can improve translation quality because the decoder can benefit from the improved language model estimates concerning potential output translations [11]. Many researchers have studied language model adaptation as well as translation model adaptation. Translation model and language model adaptation are usually used in domain adaptation for SMT.

Language model adaptation has been widely used in speech recognition [8]. In recent years, language model adaptation has also been studied for SMT. Bulyko et al. [15] explored discriminative estimation of language model weights by directly optimizing MT performance measures such as BLEU. Their experiments indicated about a 0.4 BLEU score improvement.

Eck et al. [35] developed a method to adapt language models using information retrieval methods for SMT. In their approach, they first translated input sentences with a general language model, then used these translated sentences to retrieve the most similar documents from the web applying cross language information retrieval techniques. Later these documents extracted from the web were used to build an adapted language model and then the documents were re-translated with the adapted language model.

Hildebrand et al. [51] also applied information retrieval techniques to select sentence pairs from the training corpus that are relevant to the test sentences. Both the language and the translation models are retrained on the extracted data.

Several studies investigated mixture model adaptation for both translation models and language models in SMT. Foster and Kuhn [40] investigated two basic settings: cross-domain adaptation and dynamic adaptation. In cross-domain adaptation a small sample of parallel in-domain text is assumed and in dynamic adaptation only the current input source text is considered. Adaptation relies on mixture models estimated on the training data through some unsupervised clustering methods. Given available adaptation data, mixture weights are re-estimated ad-hoc.

Semi-supervised learning has been previously applied to improve word alignments. In [24] mixture models are employed to adapt a word alignment model to in-domain parallel data. Callison-Burch et al. [18] trained a generative model for word alignment using unsupervised learning on parallel text. Also another model is trained on a small amount of hand-annotated word alignment data. A mixture model provides a probability for word alignment. Their experimental results indicate that assigning a large weight on the model trained with labeled data performs best.

Koehn and Schroeder [69] investigated different adaptation methods for SMT. They applied cross-domain adaptation techniques in a phrase-based SMT system trained on the Europarl task, in order to translate news commentaries, from French to English. They used linear interpolation techniques to exploit a small portion of in-domain bilingual data to adapt the Europarl language model and translation models. Their experiments indicate an absolute improvement of more than 1 point on their BLEU score.

Munteanu and Marcu [87] automatically extracted in-domain bilingual sentence pairs from comparable corpora in order to enlarge the in-domain bilingual corpus. They presented a novel method that uses a maximum entropy classifier that, given a pair of sentences, can reliably determine whether or not they are translations of each other in order to extract parallel data from large Chinese, Arabic, and English non-parallel newspaper corpora. They evaluated the quality of the extracted data by showing that it improves the performance of a state-of-the-art SMT system.

Callison-Burch [17] applied co-training to MT. This approach requires several source languages which are sentence-aligned with each other and all translate into the same target language. One language pair creates data for another language pair and can be naturally used in a Blum and Mitchell [10]-style co-training algorithm. Experiments on the EuroParl corpus show a decrease in WER.

Self-training for SMT was proposed in [127]. Sarkar et al. [111] proposed several elaborate adaptation methods relying on additional bilingual data synthesized from the development or test sets. They explored transductive learning for SMT, where source language corpora are used to train the models. They repeatedly translated source sentences from the development set and test set. Then the generated translations were used to improve the performance of the SMT system. They presented detailed experimental evaluations on the French–English EuroParl data set and on data from the NIST Chinese–English large data track which showed a significant improvement in translation quality on both datasets.

5.1.2 Our Semi-supervised Approach to Bengali

As mentioned in chapter 3, since a sufficient amount of bilingual parallel data between Bengali and English for SMT is not publicly available, we are exploring the use of semi-supervised techniques like self-training in SMT. We have access to approximately eleven thousand parallel sentences between Bengali and English provided by the Linguistic Data Consortium (LDC) Corpus Catalog[1]. The LDC corpus contains newswire text from the BBC Asian Network website and other South Asian language websites (eg. Bengalnet). We also have a large monolingual Bengali dataset which contains more than one million sentences. The monolingual corpus was provided by the Center for Research on Bangla Language Processing, BRAC University, Bangladesh. The corpus was built by collecting text from the Prothom Alo newspaper website and contains all the news available for the year of 2005 (from 1st January to 31st December), including magazines and periodicals. There are 18,067,470 word tokens and 386,639 distinct word types in this corpus.

We are proposing several self-training techniques to effectively use this large monolingual corpus (from the source language) in our experiments in order to improve translation accuracy. We propose several sentence selection strategies to select sentences from a large monolingual Bengali corpus, which are briefly discussed below along with the baseline system where sentences are chosen randomly.

Baseline Approach

In our baseline system the initial MT system is trained on a bilingual corpus L and we randomly select k sentences from a large monolingual corpus U. We translate these randomly selected sentences with our initial MT system $M_{B \to E}$ and denote these sentences along with their translation as U^+. Then we retrain the SMT system on $L \cup U^+$ and use the resulting model to decode the test set. We also remove these k randomly selected sentences from U. This process is continued iteratively until a certain level of translation quality, which in our case is measured by the BLEU score, is met. Below in algorithm 4, we describe the baseline algorithm.

[1] LDC Catalog No.: LDC2008E29.

Algorithm 4 Baseline Algorithm Semi-supervised SMT

```
Given bilingual corpus $L$, and monolingual corpus $U$.
$M_{B \to E} = \textbf{train}(L, \emptyset)$
for $t = 1, 2, ...$ till certain level of translation quality is reached do
    Randomly select $k$ sentence pairs from $U$
    $U^{+} = \textbf{translate}(k, M_{B \to E})$
    $M_{B \to E} = \textbf{train}(L, U^{+})$
    Remove the $k$ sentences from $U$
    Evaluate the performance on the test set $T$
end for
```

Figure 5.1 illustrates our overall baseline system. The baseline SMT system consists of a translation model, language model and the decoder. The translation model is used for initial training of the bilingual corpus and retraining with additional new sentences in each iterative step. The decoder is used to translate randomly selected sentences from the monolingual data in each iterative step and translate test data for evaluation.

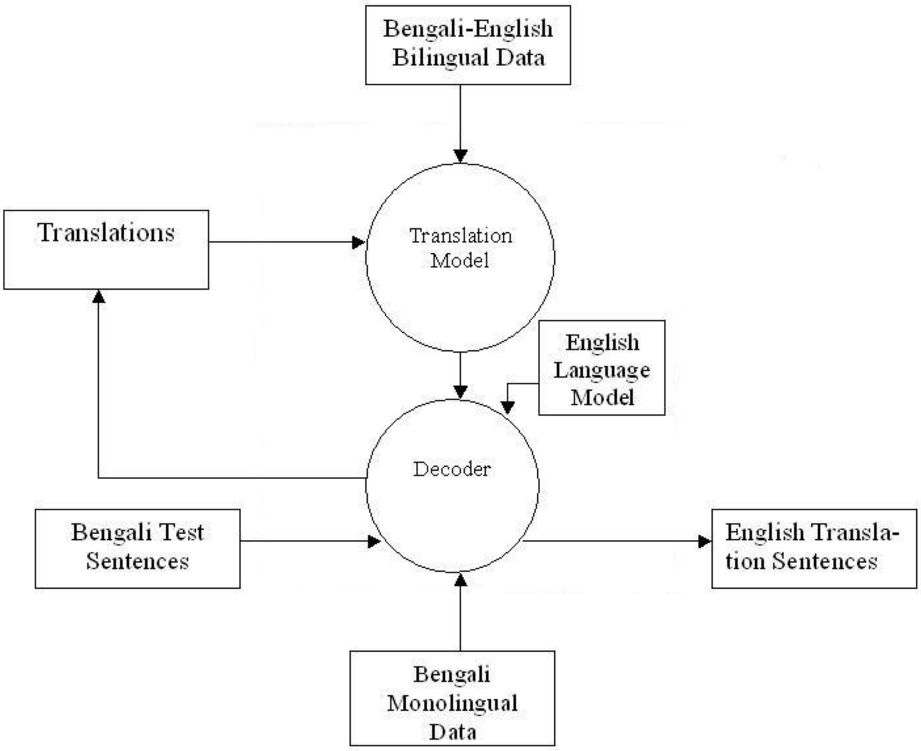


Figure 5.1: Baseline

Reverse Model Approach

Our first sentence selection approach uses the reverse translation model to rank all sentences in the monolingual corpus U based on their BLEU score and only select sentences which have higher BLEU score. Mainly we want to select those sentences from the monolingual corpus U for which our MT system can generate good translations. In order to obtain a BLEU score for sentences in the monolingual corpus U we used the reverse translation model. While a translation system $M_{B \to E}$ is built from language B to language E, we also build a translation system in the reverse direction $M_{E \to B}$. To measure the BLEU score of all monolingual sentences B from monolingual corpus U, we translate them to English sentences E by $M_{B \to E}$ and then project the translation back to Bengali using $M_{E \to B}$. We denote this reconstructed version of the original Bengali sentences by $\acute{B}$. We then use B as the reference translation to obtain the BLEU score for sentences $\acute{B}$. In algorithm 5 we describe the reverse model sentence selection algorithm.

Algorithm 5 Reverse Model sentence selection Algorithm

```
Given bilingual corpus $L$, and monolingual corpus $U$.
$M_{B \to E} = \mathbf{train}(L, \emptyset)$
$M_{E \to B} = \mathbf{train}(L, \emptyset)$
for $t = 1, 2, ...$ till certain level of translation quality is reached do
    $U^{+} : (B, E) = \mathbf{translate}(U, M_{B \to E})$
    $U^{*} : (\acute{B}, E) = \mathbf{translate}(E, M_{E \to B})$
    Use $B$ and $\acute{B}$ to rank all sentences in $U^{+}$ based on the BLEU score
    Select $k$ sentences and their translations $\acute{k}$ from ranked $U^{+}$
    $M_{B \to E} = \mathbf{train}(L, k \cup \acute{k})$
    $M_{E \to B} = \mathbf{train}(L, k \cup \acute{k})$
    Remove the $k$ sentences from $U$
    Evaluate the performance on the test set $T$
end for
```

Figure 5.2 illustrates the reverse model approach. Here the MT system consists of two translation models- one for translation in the original direction (Bengali to English) and other in the reverse direction (English to Bengali). Both translation model are initially trained with bilingual training data and retrained with new data in each iterative step. In each iteration the monolingual data is first translated with the Bengali to English decoder and the output of the decoder is used as input for the English to Bengali decoder which basically regenerates the monolingual corpus known as reverse translation. Then the quality of the reverse translation can be evaluated using monolingual data as the reference. The sentences with higher BLEU score are translated and added with bilingual training data to retrain both translation models.

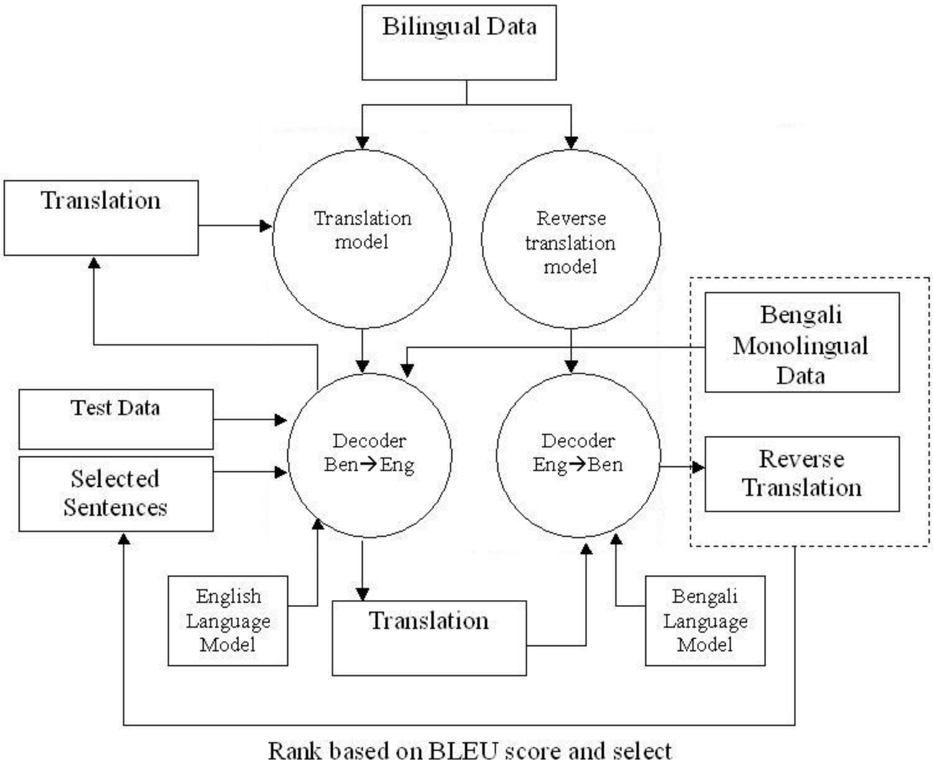


Figure 5.2: Reverse model

Frequent Word Model

Our next sentence selection approach uses statistics from the training corpus L for sentence selection from the monolingual corpus U. In this approach we first find the most frequent words in the training corpus L. We call them seed words. Then we filter the seed words based on their confidence score, which reflects how confidently we can predict their translation. Seed words with confidence scores lower than a certain threshold values are removed. Then we use these remaining seed words to select sentences from the monolingual corpus U and remove selected sentences from U. Next we look for the most frequent words other than the initial seed words in the selected sentences to be used for the next iteration as new seed words. We translate these selected sentences and add them to the training corpus L. After that we re-train the system with the new training data. In the next iteration we select new sentences from the monolingual corpus U using the new seed words and repeat the steps. We keep on repeating the steps until no more new seed words are available. In each iteration we monitor the performance on the test set T. Below in algorithm 6, we describe the frequent word sentence selection procedure.

Algorithm 6 Frequent word sentence selection Algorithm

```
Given bilingual corpus $L$, and monolingual corpus $U$.
$M_{B \to E}$ = **train**$(L, \emptyset)$
$S$ = **select_seed**$(L)$
**for** all $s$ in $S$ **do**
  **if** score(s) > threshold **then**
    $S^+ = S^+ \cup s$
  **end if**
**end for**
**while** $S^+ \neq \{\}$ **do**
  Select $k$ sentences from $U$ based on $S^+$
  U = U - K
  $U^+$ = **translate**$(k, M_{B \to E})$
  $S$ = **select_seed**$(U^+)$
  **for** all $s$ in $S$ **do**
    **if** score(s) > threshold **then**
      $S^+ = S^+ \cup s$
    **end if**
  **end for**
  $M_{B \to E}$ = **train**$(L, U^+)$
  Evaluate the performance on the test set $T$
**end while**
```

Figure 5.3 illustrates the frequent word model approach. The MT system consists of a translation model, decoder and language model. The translation model is initially trained on the bilingual training data and retrained with new data in each iterative step. Frequent words also known as seed words are selected from bilingual training data and are used to select sentences from the monolingual data. The decoder is used to translate the selected sentences from monolingual data and output of the decoder is used to retrain the translation model again. The decoder is also used to translate test data from evaluation purposes.

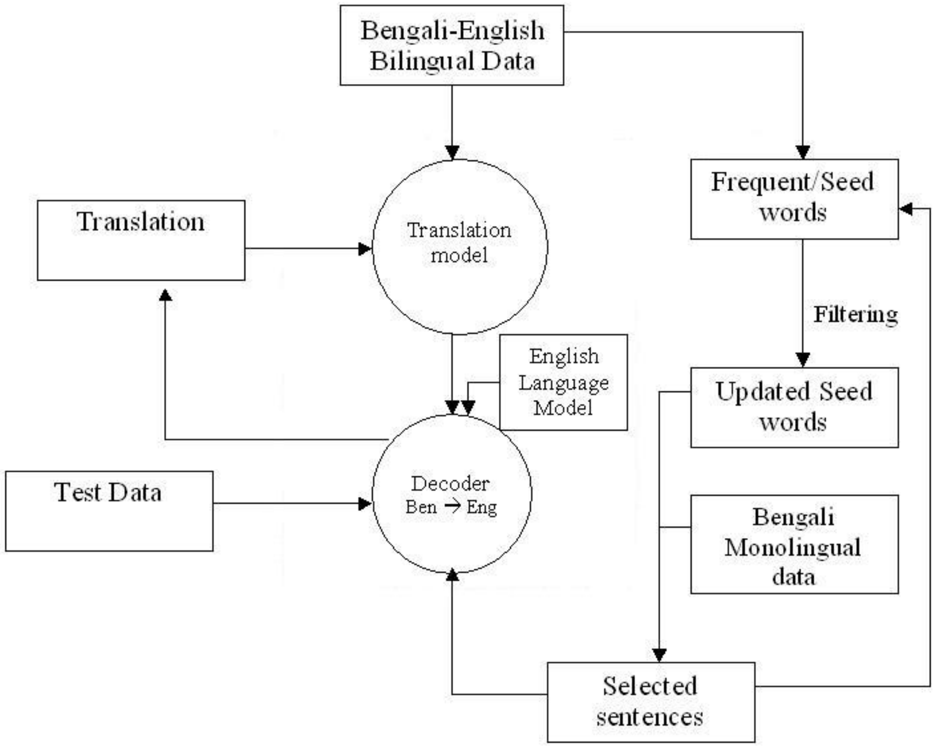


Figure 5.3: Frequent word model

5.1.3 Semi-supervised Learning Setup

We conducted semi-supervised learning experiments for Bengali SMT using the Portage[2] [128] SMT system. Similar to Moses, the models (or features) which are employed by the decoder in Portage are: (a) several phrase table(s), which model the translation direction $p(f|e)$, (b) one or several n-gram language model(s) trained with the SRILM toolkit [121]; in the experiments reported here, we used a trigram model on EuroParl, (c) a distortion model which assigns a penalty based on the number of source words which are skipped when generating a new target phrase, and (d) a word penalty. These different models are combined log-linearly. Their weights are optimized with respect to BLEU score using the algorithm described in [95] using the same development corpus provided by the LDC.

Initially we trained the translation model on the training set of 11000 sentences provided

[2] The reason for using the Portage in some of the experiments was due to the fact it was a joint collaboration work [46] with some other researchers and they preferred the Portage SMT system.

by the Linguistic Data Consortium. Then we select sentences for a Bengali monolingual dataset using one of our three approaches either random, reverse or frequent word model. In the random approach we just randomly select sentences from the monolingual dataset. In the frequent word model approach, we selects sentences from the monolingual dataset based on the most frequent words in the training data. In the reversed model approach we select sentences from the monolingual dataset that have the highest BLEU score obtained using the reversed translation model. We select 500 sentences in each iteration of the semi-supervised loop using these approaches. Then we translate selected sentences using our initial translation model and add these sentences together with their translation to our training data and retrain the system using the new dataset. Then in the next iteration we again select sentences from our monolingual dataset using one of our approaches and add them to training data after translating them with the new translation model. We continue this iterative process for a certain number of iterations and monitor the translation performance in each iteration.

5.1.4 Semi-supervised Learning Results

We applied the semi-supervised learning framework to the problem of Bengali-English SMT. In order to evaluate the effectiveness of our sentence selection strategies we tested our approaches on English-French language pair. The main reason for applying our approaches to a different language pair(English-French) is to demonstrate that our approaches are language independent and can be applied to any language pair with limited resources. Our results in Table 5.1 indicate that our reverse model approach and frequent word model out performs strong random baseline approach. For the semi-supervised learning framework, we conducted experiments on Portage instead of the Moses MT system on both language pairs.

Iterations	Random Baseline	Reverse Model	Frequent-Word Model
1	5.47	5.47	5.47
2	5.57	5.63	5.61
3	5.79	5.69	5.70
4	5.66	5.67	5.63
5	5.73	5.82	5.82
6	5.76	5.89	5.81
7	5.74	6.05	5.88
8	5.75	6.08	5.89

Table 5.1: Impact of semi-supervised learning approaches on Bengali-English SMT in BLEU score

The reverse model approach outperformed all other approaches for Bengali-English SMT however, in the case of French-English SMT, the frequent word model outperformed all other approaches. The reverse model performs better than the frequent word model for Bengali because Bengali has a rich morphology(a lot of words are inflected) so a frequent word model which is based on word frequency does not perform that well. This is not the case when translating from French to English since French and English are quite similar in their structure and grammar.

Iterations	Random Baseline	Reverse Model	Frequent-Word Model
1	13.60	13.60	13.60
2	13.61	13.61	13.63
3	13.75	13.71	13.70
4	13.82	13.80	13.93
5	13.85	13.91	13.99
6	13.90	13.94	14.01
7	13.92	14.01	14.07
8	13.93	14.03	14.17

Table 5.2: Impact of semi-supervised approaches on French-English SMT in BLEU score

The two graphs in figure 5.4 and 5.5 show the BLEU score for all the approaches for semi-supervised learning for both language pairs. We presented the results in graphs too because for iterative approaches graphs are better than tables to demonstrate how each approach performs in each iterative step.

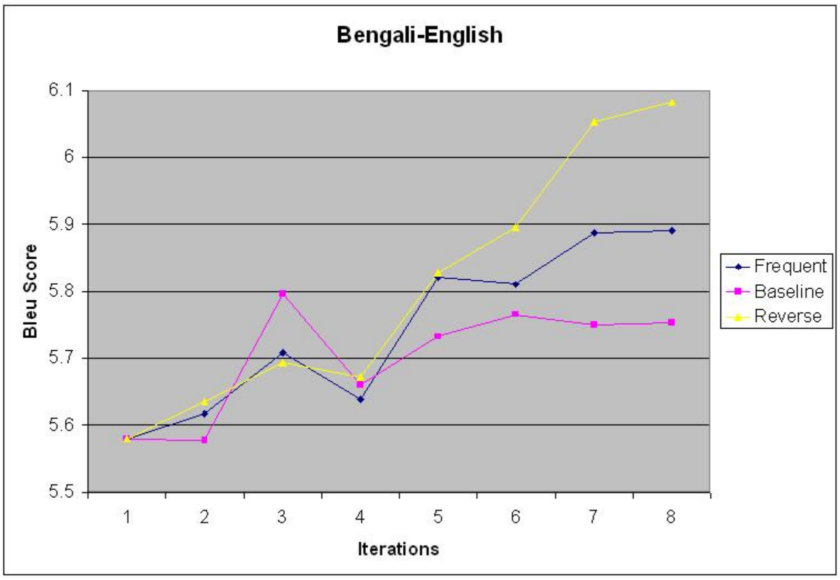


Figure 5.4: Impact of semi-supervised approaches on Bengali-English SMT

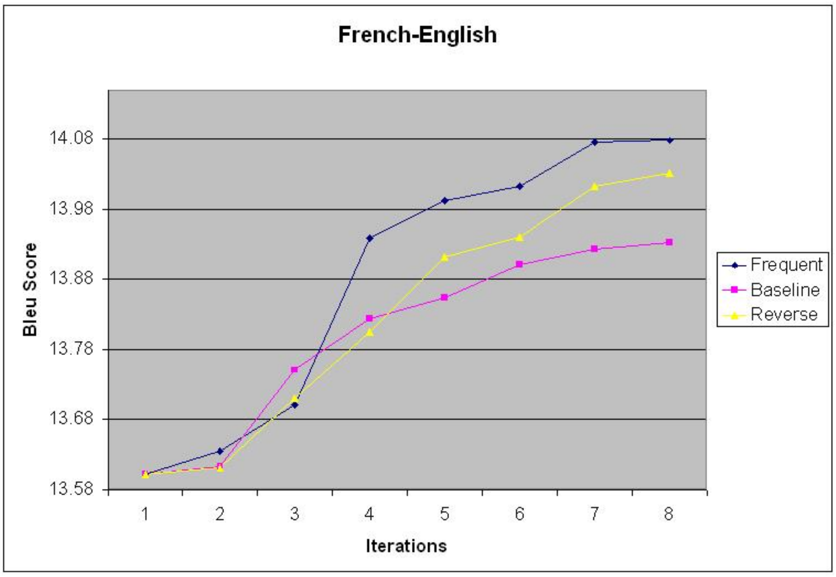


Figure 5.5: Impact of semi-supervised approaches on French-English SMT

5.2 Active Learning

Active learning(AL) is an emerging area in machine learning that explores methods that rely on actively participating in the collection of training examples rather than random sampling. In AL, a learner selects as few instances as possible to be labelled by a labeller and iteratively trains itself with the new examples selected. One of the goals of active learning is to reduce the number of supervised training examples needed to achieve a given level of performance. Also in the case where limited amount of training examples are available, to add most useful examples to the training data which can improve the performance.

Supervised learning strategies require a large set of labeled instances to perform well. In many applications, unlabeled instances may be abundant but obtaining labels for these instances could be expensive and time-consuming. AL was introduced to reduce the total cost of labeling. The process of collecting the most useful examples for training an MT system is an active learning task, as a learner can be used to select these examples.

5.2.1 Active Learning Techniques

AL systems may construct their own examples, request certain types of examples, or determine which unsupervised example are most useful for labeling. Due to the availability of an abundant amount of text and the need to annotate only the most informative sentences, the AL approach known as selective sampling [25], is particularly attractive in natural-language learning.

In selective sampling, learning begins with a small pool of annotated examples and a large pool of unannotated examples, and the learner attempts to choose the most informative additional examples for annotation. Existing work in this area has emphasized on two approaches:

1) certainty-based methods
2) committee-based methods

Certainty-based methods: In the certainty-based paradigm [74], a system is trained on a small number of annotated examples to learn an initial classifier. Then, the system examines the unannotated examples, and attaches certainties to the predicted annotation of those examples. A predefined amount of examples with the lowest certainties are then presented to the user for annotation and retraining. Many methods for attaching certainties have been used, but the methods typically estimate the probability that a classifier consistent with the prior training data will classify a new example correctly.

Committee-based methods: In the committee-based paradigm ([42], [76], [30], [25]), a diverse committee of classifiers is created, again from a small number of annotated examples. Then, each committee member labels additional examples. The examples whose annotation results in the most disagreement amongst the committee members are presented to the user for annotation and retraining. A diverse committee, consistent with the prior training data, will produce the highest disagreement on examples whose label is most uncertain with respect to the possible classifiers that could be obtained by training on that data. The density-weighted sampling strategy is also very common and is based on the idea that informative instances are those that are uncertain and representative of the input distribution.

For many language learning tasks, annotation is particularly time-consuming since it requires specifying a complex output rather than just a category label, so reducing the number of training examples required can greatly increase the utility of learning. An increasing number of researchers are successfully applying machine learning to natural language processing. However, only a few have utilized active learning techniques. Active learning, as a standard method has been applied to a variety of problems in natural language processing such as parsing ([123], [54]), automatic speech recognition [57], part of speech tagging [30], text categorization ([74],[76]), Named-Entity Recognition [112], and Word-sense disambiguation [21]. However, little work has been done in using these techniques to improve machine translation.

There has been very little work published on active learning for SMT for low-density/low-resource languages. Callison-burch [16] in his Ph.D. proposal lays out the promise of AL for SMT and proposes some algorithms. However no experimental results were reported for his approaches.

There is work on sampling sentence pairs for SMT ([60], [34]) but the goal has been to limit the amount of training data in order to reduce the memory footprint of the SMT decoder. Eck et al. [34] used a weighting scheme to sort sentences based on the frequency of unseen n-grams. After sorting they selected smaller training corpora and showed that systems trained on much less training data achieve a very competitive performance compared to baseline systems, which were trained on all available training data. They also proposed a second approach to rank sentences based on TF-IDF (term frequency–inverse document frequency) which is a widely used similarity measure in information retrieval. The TF-IDF approach did not show improvements over the other approach. They evaluated the system against a weak baseline that selected sentences based on the original order of sentences in the training corpus. Usually in such a baseline, adjacent sentences tend to be related in topic and only a few new words are added in every iteration. A random selector might have been a better baseline.

Gangadharaiah et al. [103] proposed using AL strategies to sample the most informative sentence pairs. While more data is always useful, a large training corpus can slow down an MT system. They used a pool based strategy to selectively sample the huge corpus to obtain a sub-corpus of most informative sentence pairs. Their approach outperformed a random selector and also a previously used sampling strategy [34] in an EBMT framework by about one BLEU point.

Kato and Barnard [59] implement an AL system for SMT for language pairs with limited resources (En-Xhosa, En-Zulu, En-Setswana and En-Afrikaans), but the experiments are on a very small simulated data set. The only feature used is the confidence score for sentence selection in the SMT system.

Haffari and Sarkar [47] introduced an AL task of adding a new language to an existing multilingual set of parallel text and constructing high quality MT systems, from each language in the collection into this new target language. They showed that adding a new language using AL to the EuroParl corpus provides a significant improvement in translation quality compared to a random sentence selection baseline.

5.2.2 Our Active Learning Approach to Bengali SMT

In this section we provide an experimental study of AL for Bengali-English SMT. Specifically, we use AL to improve quality of a phrase-based Bengali-English SMT system since a limited amount of bilingual data is available for the language pair.

In order to improve or adapt an SMT system an obvious strategy is to create or add more new bilingual data to the existing bilingual corpora. However, just randomly translating text and adding to the bilingual corpora might not always benefit SMT systems since new translated sentences might be similar to the existing bilingual corpora and might not contribute a lot of new phrases to the SMT system. Selective sampling of sentences for AL will lead to a parallel corpus where each sentence does not share any phrase pairs with the existing bilingual corpora and the SMT system will benefit for the new phrase pairs.

We use a novel framework for AL. We assume a small amount of parallel text and a large amount of monolingual source language text. Using these resources, we create a large noisy parallel text which we then iteratively improve using small injections of human translations.

Starting from an SMT model trained initially on bilingual data, the problem is to minimize the human effort involved with translating new sentences which will be added to the training data to make the *retrained* SMT model achieve a certain level of performance. Thus, given a bitext $L := \{(\mathbf{f}_i, \mathbf{e}_i)\}$ and a monolingual source text $U := \{\mathbf{f}_j\}$, the goal is to select a subset of highly informative sentences from U to present to a human expert for translation. Highly informative sentences are those which, together with their translations, help the retrained SMT system *quickly* reach a certain level of translation quality.

Algorithm 7 describes the experimental setup we propose for AL. We train our initial MT system on the bilingual corpus L, and use it to translate *all* monolingual sentences in U. We denote sentences in U together with their translations as U^+ (line 4 of Algorithm 7). Then we retrain the SMT system on $L \cup U^+$ and use the resulting model to decode the test set. Afterwards, we select and remove a subset of highly informative sentences from U, and add those sentences together with their human-provided translations to L. This process is continued iteratively until a certain level of translation quality, which in our case is measured by the BLEU score, is met. In the baseline, against which we compare the sentence selection methods, the sentences are chosen *randomly.* When (re-)training the model, two phrase tables are learned: one from L and the other one from U^+.

The setup in Algorithm 7 helps us to investigate how to maximally take advantage of human effort (for sentence translation) when learning an SMT model from the available data, that includes bilingual and monolingual text. $M_{F \to E}$ in Algorithm 7 denotes a MT system that translates from language F to E.

Algorithm 7 AL-SMT

```
Given bilingual corpus $L$, and monolingual corpus $U$.
$M_{F \rightarrow E}$ = **train**$(L, \emptyset)$
**for** $t = 1, 2, ...$ **do**
    $U^{+}$ = **translate**$(U, M_{F \rightarrow E})$
    Select $k$ sentence pairs from $U^{+}$, and ask a human for their *true* translations.
    Remove the $k$ sentences from $U$, and add the $k$ sentence pairs (translated by human) to $L$
    $M_{F \rightarrow E}$ = **train**$(L, U^{+})$
    Evaluate the performance on the test set $T$
**end for**
```

Figure 5.6 illustrates the overall AL setting.

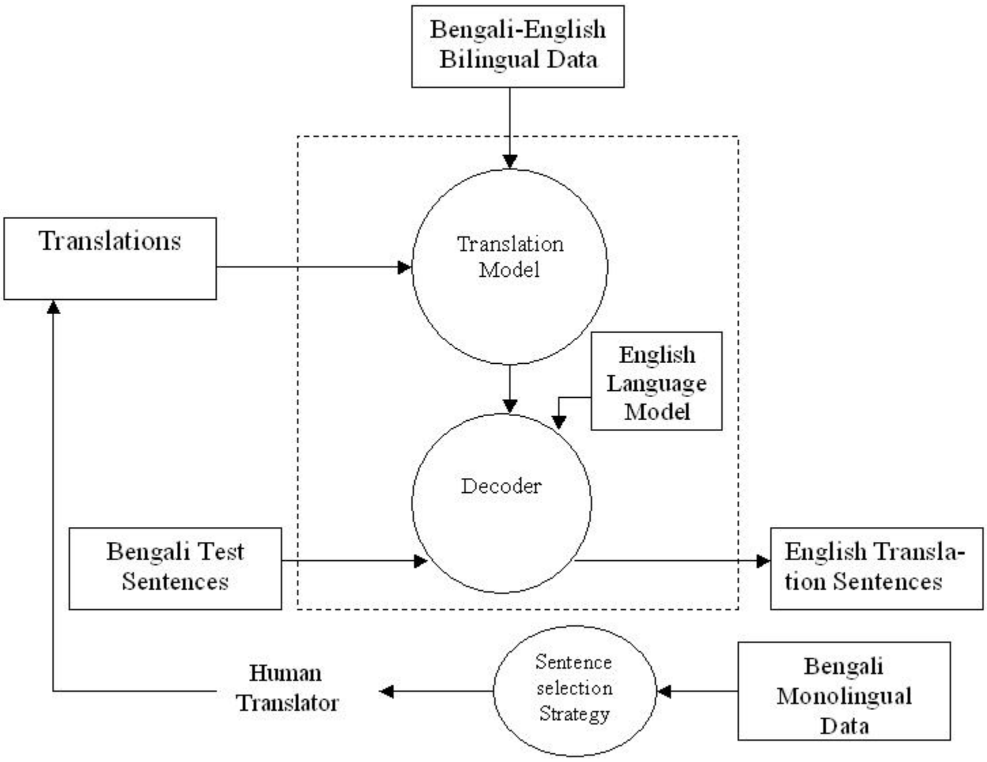


Figure 5.6: Active learning setting

5.2.3 Sentence Selection Strategies

Below we discuss several sentence selection strategies proposed by [46] and used in our AL scenario for Bengali-English SMT.

Geometric-Phrase and Arithmatic-Phrase

The more frequent a phrase is in the *unlabeled* data, the more important it is to know its translation; since it is more likely to occur in the test data (especially when the test data is in-domain with respect to unlabeled data). The more frequent a phrase is in the *labeled* data, the more unimportant it is; since probably we have observed most of its translations.

Based on the above observations, we measure the importance score of a sentence as:

$$\phi_g^p(s) := \Big[\prod_{x \in X_s^p} \frac{P(x|U)}{P(x|L)} \Big]^{\frac{1}{|X_s^p|}} \tag{5.1}$$

where X_s^p is the set of possible phrases that sentence s can offer, and $P(x|\mathcal{D})$ is the probability of observing x in the data $\mathcal{D}$: $P(x|\mathcal{D}) = \frac{Count(x)+\epsilon}{\sum_{x \in X_{\mathcal{D}}^p} Count(x)+\epsilon}$. The score (5.1) is the averaged *probability ratio* of the set of candidate phrases, i.e. the probability of the candidate phrases under a probabilistic phrase model based on U divided by that based on L. In addition to the geometric average in (5.1), we may also consider the arithmetic average score:

$$\phi_a^p(s) := \frac{1}{|X_s^p|} \sum_{x \in X_s^p} \frac{P(x|U)}{P(x|L)} \tag{5.2}$$

Note that (5.1) can be re-written as $\frac{1}{|X_s^p|} \sum_{x \in X_s^p} \log \frac{P(x|U)}{P(x|L)}$ in the logarithm space, which is similar to (5.2) with the difference of additional log.

Geometric n-gram and Arithmatic n-gram

As an alternative to phrases, we consider n-grams as basic units of generalization. The resulting score is the weighted combination of the n-gram based scores:

$$\phi_g^N(s) := \sum_{n=1}^{N} \frac{w_n}{|X_s^n|} \sum_{x \in X_s^n} \log \frac{P(x|U,n)}{P(x|L,n)} \tag{5.3}$$

where X_s^n denotes n-grams in the sentence s, and $P(x|\mathcal{D},n)$ is the probability of x in the set of n-grams in $\mathcal{D}$. The weights w_n adjust the importance of the scores of n-grams with different lengths. In addition to taking geometric average, we also consider the arithmetic average:

$$\phi_a^N(s) := \sum_{n=1}^{N} \frac{w_n}{|X_s^n|} \sum_{x \in X_s^n} \frac{P(x|U,n)}{P(x|L,n)} \tag{5.4}$$

As a special case when $N = 1$, the score motivates selecting sentences which increase the number of unique words with new words appearing with higher frequency in U than L.

5.2.4 Active Learning Setup

We applied active learning to the Bengali-English SMT task to create a larger Bengali-English parallel text resource. Similar to the semi-supervised learning approach, we train our initial translation model on the training set of 11000 sentence provided by the Linguistic Data Consortium. Then we select sentences for a Bengali monolingual dataset using one of the four sentence selection strategies, which are Geometric-Phrase, Arithmetic-Phrase, Geometric n-gram and Arithmetic n-gram. A user participated in the AL loop, translating 100 sentences in each iteration. In each iteration we added the selected sentences and their translation to our training data and retrained the model. We continued this iterative process for 6 iterations and monitored the translation perform in each iteration. Also as part of active learning loop we created a small parallel corpora of 3000 new sentences between Bengali and English. Since each iteration in AL loop is very time consuming due to the manual translation we only translated 100 sentences in each iteration.

5.2.5 Active Learning Results

Our experimental results show that adding more human translation does not always result in better translation performance. This is likely due to the fact that the translator in the AL loop was not the same as the original translator for the labeled data. The results are shown in below table 5.3. Geom 4-gram and Geom phrase are the features that prove most useful in extracting useful sentences for the human expert to translate.

Iterations	Random Baseline	Geometric 4-gram	Geometric Phrase
1	5.42	5.42	5.42
2	5.17	5.34	5.14
3	5.25	5.42	5.07
4	5.40	5.58	5.23
5	5.49	5.65	5.40
6	5.46	5.66	5.62

Table 5.3: Impact of active learning approaches on Bengali-English SMT in BLEU score

Below the graph 5.7 also represents the BLEU score for random, geometric 4-gram and geometric phrase sentence selection strategy in the AL setting for Bengali-English SMT. WE see that two of the graphs have leveled off after 5 iterations. All dropped significantly in the initial iterations before recovering.

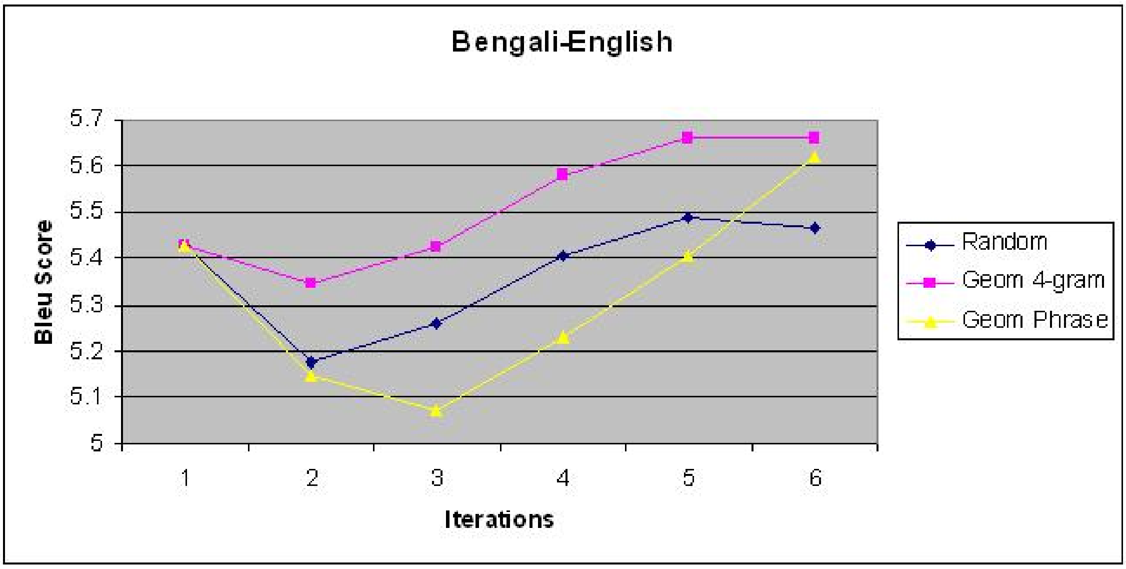


Figure 5.7: Impact of active learning approaches on Bengali-English SMT

5.3 Summary

In this chapter, we provided a novel semi-supervised learning and active learning framework for SMT which utilizes both labeled and unlabeled data. Several sentence selection strategies were discussed and detailed experimental evaluations were performed on the sentence selection method. In semi-supervised settings, reversed model approach outperformed all other approaches for Bengali-English SMT and in active learning setting, geometric 4-gram and geometric phrase sentence selection strategies proved most useful.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We have discussed different approaches that can be applied to Bengali-English Statistical Machine Translation. Our work has not only provided advances in Bengali SMT, but has also provided a foundation for conducting research in Bengali SMT. We made several novel contributions starting from building different Bengali language processing tools including a morphological analyzer, tokenizer, compound splitter, preposition and transliteration handling module. We have also applied several novel approaches to Bengali-English SMT incorporating word reordering, semi-supervised and active learning. We showed that these approaches help to improve translation quality in a Bengali-English SMT system. We created a better test set for evaluating Bengali-English MT task and also provided a test set for evaluating the compound splitting module and transliteration module.

Given the lack of sufficient bilingual corpora between Bengali and English we have investigated several novel approaches in a Bengali-English SMT system. We have demonstrated that for low-density language like Bengali, these rule-based and machine learning approaches can improve translation quality. In summary, the following scientific contributions have been achieved:

1. We contributed by describing the first phrase-based SMT system from Bengali to English which incorporates transliteration, compound word and prepositional module in order to deal with limited resources.

2. We applied several word reordering approaches to the Bengali-English STM system. We reordered Bengali sentences by applying reordering rules learnt automatically. We also

applied manually predefined rules to reorder Bengali sentences and lexicalized reordering techniques. We evaluated the approaches through their impact on the BLEU score and automatic reordering approach showed a 1.4 BLEU score improvement over the baseline.

3. We proposed two semi-supervised learning techniques for sentence selection within a Bengali-English Phrase-based SMT System. We showed improvement in BLEU score for both approaches over the baseline approach.

4. We proposed several effective active learning techniques for sentence selection from a pool of untranslated sentences, for which we asked human experts to provide translations. We also contributed new parallel corpora through the active learning loop.

5. We contributed a better test set with three reference test sets for evaluation of translation quality between Bengali and English SMT systems. We also proposed a new manual evaluation approach for evaluating SMT output that requires less human effort than other approaches.

6.2 Future Work

In this section we outline some future directions for Bengali language research based on the research outlined in this thesis.

6.2.1 Applying Hierarchical Phrases-based Statistical Machine Translation for Bengali English SMT

A Hierarchical phrase-based SMT model [23] uses hierarchical phrases — phrases that contain subphrases. The model is formally a synchronous context-free grammar [3] but is learned from a parallel text without any syntactic annotations. Thus it can be seen as combining fundamental ideas from both syntax-based translation and phrase-based translation. In our experiments we only used a phrase-based SMT system. Since Bengali and English have differences in word order, it would be worthwhile to investigate the performance of the Bengali English SMT on a Hierarchical phrases-based SMT model.

6.2.2 Development of Bengali Treebank

One of our future projects is to develop a Bengali treebank. A treebank or parsed corpus is a text corpus in which each sentence has been annotated with syntactic (tree) structure.

Treebanks can be created completely manually, where linguists annotate each sentence with syntactic structure. The degree of difficulty associated with treebank construction depends of the level of annotation detail. Treebanks can take years to build [2]. Mahmud and Khan [80] proposed an outline of a semi-automated process for developing a linguistically annotated corpus for the Bengali language which could be our starting point toward building the Bengali treebank. Having a Bengali treebank will help to build other Bengali language processing tools which will benefit Bengali MT research and Bengali NLP in general.

6.2.3 Development of Additional Bengali tools

Development of the Bengali treebank mentioned above can help in building a Bengali parser. Treebanks can be used in computational linguistics for training or testing parsers. In this thesis the word reordering approaches are based on a Bengali part of speech tagger. However, some of the successful word reordering approaches are based on parser output. So having a parser in place would benefit Bengali MT by having a better word reordering models. We may also investigate building a semantic role labeller which will benefit Bengali SMT and other applications.

6.2.4 Improving Current Modules

It would also be worthwhile to investigate how to improve the accuracy of our current transliteration, prepositional and compound word module. A starting point would be to handle different types of prepositions and compound words which are currently not being handled. Also it would be useful to explore other approaches or incorporate more resources to improve accuracy of the transliteration module.

6.2.5 Increasing Amount of Resources

While our focus has been on achieving quality translations using minimal resources, the availability of more resources could definitely improve performance. Since the bilingual corpora between Bengali and English is still not adequate for SMT systems, it is necessary to create or collect more bilingual data between Bengali and English.

Appendix A

Manual SMT Evaluation

In this appendix we provide the survey questions we used for manual SMT evaluation and the user responses for all individual questions.

A.1 Survey Questions

Below we provide all the survey questions. Here option 1 is the output of the baseline system and option 2 is output of our system.

Q1. Reference Sentence:"Euro-Asian culture and civilization convention in Paris is over." Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Culture and in the conference of concluded

Option 2: Of the euro , asia culture and paris conference concluded.

Option 3: Both similar

Q2. Reference Sentence:"African Union will lead the peace keeping initiative this time." Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Now in the peace protect the leadership.

Option 2: The african union now with the leadership of the activities of peace.

Option 3: Both similar

Q3. Reference Sentence: "All soldiers in peace keeping force have come from African countries." Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Forces peace protect all the soldiers from the countries to.

Option 2: Of the forces of peace all the soldiers came from the african countries.

Option 3: Both similar

Q4. Reference Sentence: ”They were arrested because of having links with Al-Quida.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Relationship with their there is arrested.

Option 2: There is the al-qaeda relations with their arrested.

Option 3: Both similar

Q5. Reference sentence:”The scene is very gorgeous.” Pick the option which you think is more similar to the reference sentence. Please note that none of options might be grammatically correct.

Option 1: the pictures are very.

Option 2: The very beauty.

Option 3: Both similar

Q6. Reference sentence: ”The rail is the most popular means of transport.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Exchanges railway vehicles is the most popular in the.

Option 2: Ral vehicles in the air traffic is the most popular contacts.

Option 3: Both similar

Q7. Reference sentence:”Today’s program is very good, a lot can be learned through it.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Today’s programme very good in signs of more knowledge

Option 2: Today programme very good , as a result of more knowledge available.

Option 3: Both similar

Q8. Reference sentence:”China all set to publish a list of technology for importing.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Of china will expressed list of technology.

Option 2: China has expressed the deliberations of technology will.

Option 3: Both similar

Q9. Reference sentence:”Sirajuddin said that China is the Friendly Neighbour for Malaysia” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Friendly neighboring eager china.

Option 2: Shirazudin said that china to friendly neighbour countries .

Option 3: Both similar

Q10. Reference sentence:”It was further mentioned in the report that, China’s future economic possibilities are good on the whole.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Been further mentioned in the report on the next economic between china

Option 2: The report also it is mentioned that china , for the next economic they are on the good.

Option 3: Both similar

Q11. Reference sentence: ”New Prime Minister Hania said, new Government under his leadership will not oppose discussion between Abbas and Israel.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Of the new prime minister said that his on the new government abbas and it will objects the discussion.

Option 2: The new prime minister hani said that he headed by the new government abbas and it will not objects of the discussion .

Option 3: Both similar

Q12. Reference sentence:”Japan will not solely propose expansion of UN Security Council.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Friendship nations of the security council will not proposal

Option 2: Japan the un security council will not give the proposal.

Option 3: Both similar

Q13. Reference sentence: ”At the beginning of the year he had to appear in court in connection with a scandal.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: At the current year a court in front of the him to.

Option 2: At the current year a him to issues in front of the court.

Option 3: Both similar

Q14. Reference sentence: ”To prevent such discrimiation, the European Union created specific anti-discriminatory guidelines in 2000.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Formed not so that the two thousand in the union for some for.

Option 2: This was not aware of the type of so that the european union for the two thousand in dealing with some of them.

Option 3: Both similar

Q15. Reference Sentence:”The employee went to the apartment and rang the doorbell.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: The workers in the bell authorities

Option 2: The workers , to the bell saw .

Option 3: Both similar

Q16. Reference Sentence:”He said the discovered relief work of three animals proves that humans were aware of the art of drawing and carving in the Paleolithic age.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: He said it will be three of the that this era of the stone pathway skilled to industry.

Option 2: He said it would be three of the evidence of that , in the courts era stone pathway to industry was skilled.

Option 3: Both similar

Q17. Reference Sentence:”It is being said that the works of art are from the time when the ancient modern Homo sapiens migrated to Europe.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: It is said a when the modern hit in Europe.

Option 2: It is said that the car when a modern homo protestors reached an agreement on europe .

Option 3: Both similar

Q18. Reference Sentence:”Today I am writing about a national park.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: Discuss a national park today.

Option 2: Is a national park discussed today.

Option 3: Both similar

Q19. Reference Sentence:”Our state of Nordrhein-Westfalen had its first National Park inaugurated in Eiffel on January 11.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: The north west of the bus preach in that we of the first of the national park was inaugurated last january.

Option 2: That is the north west of the rhein myself , that we live in the national park was the first is the last january.

Option 3: Both similar

Q20. Reference sentence:”In his inaugural speech, he said that the park is a national treasure.” Pick the option which you think is more similar to the reference sentence. Please note that none of the options might be grammatically correct.

Option 1: He his inaugural speech of the national park is a national

Option 2: He expressed his inaugural speech said , this national park is a national resource.

Option 2: Both similar

A.2 User Results

In the table A.1 we provide the responses of the survey participants for each survey question.

Question	Baseline	Our System	Both Similar
1	7	11	4
2	2	20	0
3	1	20	1
4	3	14	5
5	3	10	9
6	14	1	7
7	6	13	3
8	5	9	8
9	2	17	3
10	1	19	2
11	2	18	2
12	1	16	5
13	3	10	9
14	2	11	9
15	5	3	14
16	6	8	8
17	12	4	6
18	13	2	7
19	10	4	8
20	0	20	2

Table A.1: Survey results

Appendix B

Bengali Language Processing Tools

In this appendix we discuss some of the language processing tools we developed for the Bengali language such as a tokenizer, sentence segmentizer, Bengali script Identifier. We also created a Bengali lexicon which is part of some of the language processing tools.

B.1 Bengali Tokenizer

Usually tokenizer is used in a preprocessing step in NLP applications. The tokenizer identifies tokens. The entity word is one kind of token for NLP, the most basic one. Tokenization is the process of breaking a stream of text up into meaningful elements or tokens. Besides identifying and tokenizing common sentence delimiters such as exclamation mark or question mark, the Bengali tokenizer needs to handle some special delimiters for the Bengali language such as Dari (u09F7). The unicode of some of the delimiters Bengali tokenizer handles are (u0964), (u0965), (u09F7), (u09FB). We used Python to develop our Bengali tokenizer.

B.2 Sentence Segmenter

A sentence segmenter is used to detect sentence boundaries. We used a sentence segmenter in a preprocessing step of our Bengali SMT system. We used a Perl script to develop our sentence segmentizer. This Perl script takes a text file as standard input and splits it up so that each sentence is on a separate line.

The script determines the place of a sentence boundary on the basis of sentences delimiters such as exclamation marks or question marks and special Bengali delimiters such as danda(u0964), double danda (u0965), and dari(u09F7).

B.3 Bengali Script Identifier

We developed a special tool named Bengali script identifier which identifies Bengali script in a file which is mixed with Bengali and English. We used the script in a post-processing step of our SMT system to identify the untranslated Bengali words to apply our transliteration and prepositional module. We used Python to develop the script.

B.4 Baseline Moses Script

Below we describe the baseline script for Moses SMT system.

```
# tokenize training files #
export PATH=$PATH:.:/cs/packages/moses/bin/scripts
head -$TRAINSIZE allcorpus.bn | bengali-tokenizer.perl > corpus/allcorpus.tok.bn
head -$TRAINSIZE allcorpus.en | tokenizer.perl -l en > corpus/allcorpus.tok.en
# cleanup #
export PATH=$PATH:.:/cs/packages/moses/bin/moses-scripts/training
clean-corpus-n.perl corpus/allcorpus.tok bn en corpus/allcorpus.clean 1 40
# lowercase #
export PATH=$PATH:.:/cs/packages/moses/bin/scripts
lowercase.perl < corpus/allcorpus.clean.en > corpus/allcorpus.lowercased.en
cp corpus/allcorpus.clean.bn corpus/allcorpus.lowercased.bn
# Language Model #
export PATH=$PATH:.:/cs/packages/srilm/bin/i686-m64
ngram-count -order 3 -interpolate -kndiscount -text lm/allcorpus.out -lm lm/allcorpus.lm
# Training #
export PATH=$PATH:.:/cs/packages/moses/bin/moses-scripts/training
export SCRIPTS-ROOTDIR=/cs/packages/moses/bin/moses-scripts
train-factored-phrase-model.perl -root-dir . –corpus corpus/allcorpus.lowercased -f bn -e
en -alignment grow-diag-final-and -lm 0:5:/cs/maxim/moses-expt/baseline/working-dir/
```

```
lm/allcorpus3.lm:0
# Tuning #
export PATH=$PATH:.:/cs/packages/moses/bin/scripts
head -$DEVSIZE dev-corpus.bn | bengali-tokenizer.perl > tuning/input
head -$DEVSIZE dev-corpus.en | tokenizer.perl -l en > tuning/reference.tok
lowercase.perl < tuning/reference.tok > tuning/reference
export PATH=$PATH:.:/cs/packages/moses/scripts/training
export SCRIPTS-ROOTDIR=/cs/packages/moses/scripts/
mert-moses-new.pl tuning/input tuning/reference /cs/packages/moses/moses-cmd/src/moses
model/moses.ini –working-dir /cs/maxim/moses-expt/baseline/working-dir/tuning
–mertdir /cs/packages/moses/mert
export PATH=$PATH:.:/cs/packages/moses/bin/scripts
reuse-weights.perl tuning/moses.ini < model/moses.ini > tuning/moses.weight-reused.ini
# Decoding #
export PATH=$PATH:.:/cs/packages/moses/bin/scripts
head -$EVALSIZE test-corpus.bn | bengali-tokenizer.perl > corpus/eval.lowercased.txt
export PATH=$PATH:.:/cs/packages/moses/moses-cmd/src
moses -config tuning/moses.weight-reused.ini -mbr -drop-unknown -input-file
corpus/eval.lowercased.txt > corpus/english.output
```

Bibliography

[1] N. AbdulJaleel and L. S. Larkey. Statistical transliteration for english-arabic cross language information retrieval. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management, pages: 139-146*, New Orleans, LA, 2003.

[2] A. Abeillé. Treebanks. building and using parsed corpora. In *Series: Text, Speech and Language Technology*, Vol. 20. Abeillé, A. (Ed.), 2003.

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

[4] Y. Al-onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. Statistical machine translation. In *Final Report, JHU Summer Workshop*, 1999.

[5] Y. Al-Onaizan and K. Knight. Machine transliteration of names in arabic text. In *ACL Workshop on Computational Approaches to Semitic Languages*, 2002.

[6] H. Alshawi, S. Bangalore, and S. Douglas. Learning dependency translation models as collections of finite state head transducers. In *Computational Linguistics, 26(1):45-60*, 2000.

[7] M. Arbabi, S. M. Fischthal, V. C. Cheng, and E. Bart. Algorithms for arabic name transliteration. In *IBM Journal of Research and Development, 38(2):183-193*, 1994.

[8] M. Bacchiani and B. Roark. Unsupervised language model adaptation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages: 224-227*, 2003.

[9] A. L. Berger, S. A. D. Pietra, and V. J. Della. A maximum entropy approach to natural language processing. In *Computational Linguistics, 22(1):39-72*, 1996.

[10] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[11] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In *In Proceedings of EMNLP CoNLL*, Prague, 2007.

[12] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. D. Lafferty, and R. L. Mercer. Analysis, statistical transfer, and synthesis in machine translation. In *Proceedings of Conference on Theoretical and Methodological Issues in Machine Translation, pages: 83-100*, 1992.

[13] P. F. Brown, V. J. Pietra, S. A. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics 19: 2*, 1993.

[14] R. D. Brown. Corpus-driven splitting of compound words. In *Proceedings of the Ninth International Conference on Theoretical and Methodological Issues in Machine Translation*, 2002.

[15] I. Bulyko, S. Matsoukas, R. Schwartz, L. Nguyen, and J. Makhoul. Language model adaptation in machine translation from speech. In *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007.

[16] C. Callison-burch. Active learning for statistical machine translation. In *PhD Proposal, Edinburgh University*, 2003.

[17] C. Callison-Burch. Co-training for statistical machine translation. In *Master's thesis*, School of Informatics, University of Edinburgh, 2002.

[18] C. Callison-Burch, D. Talbot, and M. Osborne. Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, pages: 175-es*, Barcelona, Spain, July 21-26, 2004.

[19] B. Chen, M. Cettolo, and M. Federico. Reordering rules for phrase-based statistical machine translation. In *Proceeding of IWSLT 2006. pages: 182-189*, Kyoto, Japan, Nov. 2006.

[20] H. Chen, S. Huang, Y. Ding, and S. Tsai. Proper name translation in cross-language information retrieval. In *Proceedings of 17th COLING and 36th ACL, pages: 232-236*, 1998.

[21] J. Chen, A. Schein, L. Ungar, and M. Palmer. An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of HLT-NAACL*, 2006.

[22] K.H. Chen and H.H. Chen. A rule-based and mt-oriented approach to prepositional phrases attachment. In *Proceedings of the 16th International Conference on Computational Linguistics, pages: 216–221*, Copenhagen, Denmark, 1996.

[23] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL, pages: 263–270*, 2005.

[24] J. Civera and A. Juan. Word alignment quality in the ibm 2 mixture model. In *PRIS-08, pages: 93-102*, 2008.

[25] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. In *Machine Learning 15(2):201-221*, 1994.

[26] M. Collins, P. Koehn, and I. Kucerov. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages: 531-540*, Ann Arbor, Michigan, 2005.

[27] M. R. Costa-jussa and J. A. R. Fonollosa. Statistical machine reordering. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing, pages: 70–76*, Sydney, Australia, July, 2006.

[28] J. M. Crego and J. B. Marinov. Syntax-enhanced ngram-based smt. In *Proceedings of the Machine Translation Summit (MT SUMMIT XI)*, 2007.

[29] J. M. Crego and J. B. Mariño. Reordering experiments for n-gram-based smt. In *1st IEEE/ACL International Workshop on Spoken Language Technology (SLT'06), pages: 242-245*, Palm Beach (Aruba), December 2006.

[30] I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML-95, pages: 150-157*, 1995.

[31] S. Dasgupta, D. S. Hossain, A. I. Sirkar, N. Khan, and M. Khan. Morphological analysis of inflectional compound words in bangla. In *Proceedings of the 8th International Conference on Computer and Information Technology (ICCIT)*, Bangladesh, 2005.

[32] S. Dasgupta, A. Wasif, and S. Azam. An optimal way towards machine translation from english to bengali. In *Proceedings of the 7th International Conference on Computer and Information Technology (ICCIT)*, Bangladesh, 2004.

[33] E. Dura. Parsing words. In *Ph.D. thesis, Goteborg University*, Goteborg, Sweden, 1998.

[34] M. Eck, S. Vogel, and A. Waibel. Low cost portability for statistical machine translation based in n-gram frequency and tf-idf. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, 2005.

[35] M. Eck, S. Vogel, and A. Waibel. Language model adaptation for statistical machine translation based on information retrieval. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC), pages: 327–330*, Lisbon, Portugal, 2003.

[36] J. Eisner. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of ACL-03 (Companion Volume), pages: 205-208*, 2003.

[37] J. Emonds. A unified theory of syntactic categories. In *Dordrecht: Foris*, 1985.

[38] M. Federico, N. Bertoldi, and M. Cettolo. Irstlm: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech, pages: 1618-1621*, Melbourne, Australia, 2008.

[39] M. Federico and M. Cettolo. Efficient handling of n-gram language models for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation, pages: 88-95*, Prague, Czech Republic, 2007.

[40] G. Foster and R. Kuhn. Mixturemodel adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation, pages: 128–135*, Prague, Czech Republic, 2007.

[41] A. Freeman, Condon S., and Ackerman C. Cross linguistic name matching in english and arabic. In *Human Language Technology Conference of the NAACL, pages 471–478, Association for Computational Linguistics*, New York City, USA, June, 2006.

[42] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. In *Machine Learning, 28, pages: 133-168*, 1997.

[43] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, July 17-21, Sydney, Australia, 2006.

[44] D. Gildea. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, pages: 80-87*, Sapporo, Japan, 2003.

[45] J. Graehl and K. Knight. Training tree transducers. In *Proceedings NAACL-HLT*, 2004.

[46] G. Haffari, M. Roy, and A. Sarkar. Active learning for statistical phrase-based machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2009.

[47] G. Haffari and A. Sarkar. Active learning for multilingual statistical phrase based machine translation. In *the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, Singapore, 2009.

[48] J. Hajič, M. Čmejrek, B. Dorr, Y. Ding, J. Eisner, D. Gildea, T. Koo, K. Parton, G. Penn, D. Radev, and O. Rambow. Natural language generation in the context of machine translation. In *Technical report, Center for Language and Speech Processing*, Johns Hopkins University, Baltimore, Summer Workshop Final Report, 2002.

[49] M. Hearne and A. Way. Seeing the wood for the trees: Data-oriented translation. In *Proceedings of MT Summit IX*, New Orleans, September 2003.

[50] T. Hedlund, H. Keskustalo, A. Pirkola, E. Airio, and K. Jarvelin. Utaclir @ clef 2001 - effects of compound splitting and n-gram techniques. In *Second Workshop of the Cross-Language Evaluation Forum (CLEF)*, 2001.

[51] M. S. Hildebrand, M. Eck, S. Vogel, and A. Waibel. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of the 10th EAMT conference Practical applications of machine translation, pages: 133–142*, Budapest, May, 2005.

[52] W. J. Hutchins. Machine translation: A concise history. In *Chan Sin Wai (Ed.) Computer aided translation: Theory and practice*, China: Chinese University of Hong Kong, 2007.

[53] W. J. Hutchins and H. L. Somers. An introduction to machine translation. In *London: Academic Press Limited*, 1992.

[54] R. Hwa. Sample selection for statistical parsing. In *Computational Linguistics, 30(3):253–276*, 2004.

[55] R. Jackendoff. The architecture of the language. In *Cambridge, MA: MIT Press*, 1977.

[56] J. B. Johannesen and H. Hauglin. An automatic analysis of norwegian compounds. In *T. Haukioja, editor, Papers from the 16th Scandinavian Conference of Linguistics, pages: 209–220*, Turku/Åbo, Finland, 1996.

[57] T. Kamm and G. L. Meyer. Selective sampling of training data for speech recognition. In *Human Language Technology*, March 24-27, San Diego, California, 2002.

[58] M. M. Kashani, F. Popowich, and A. Sarkar. Automatic transliteration of proper nouns from arabic to english. In *Proceedings of the Second Workshop on Computational Approaches to Arabic Script-based Languages, CAASL-2. LSA*, Linguistic Institute, Stanford University. July 21-22, 2007.

[59] R.S.M. Kato and E. Barnard. Statistical translation with scarce resources: a south african case study. In *SAIEE Africa Research Journal, 98(4):136–140*, December, 2007.

[60] D. Kauchak. Contribution to research on machine translation. In *PhD Thesis*, University of California at San Diego, 2006.

[61] K. Knight and J. Graehl. Machine transliteration. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, pages: 128 - 135*, Madrid, Spain, 1997.

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

[63] P. Koehn, A. Arun, and H. Hoang. Towards better machine translation quality for the german-english language pairs. In *Proceedings of the Third Workshop on Statistical Machine Translation, pages: 139–142*, Columbus, Ohio, 2008.

[64] P. Koehn and H. Hieu. Factored translation models. In *Proceedings of EMNLP*, 2007.

[65] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstration session*, Prague, Czech Republic, June, 2007.

[66] P. Koehn and K. Knight. Feature-rich statistical translation of noun phrases. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, pages: 311 - 318*, Sapporo, Japan, 2003.

[67] P. Koehn and K. Knight. Empirical methods for compound splitting. In *Proceedings of EACL*, 2006.

[68] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, May 27-June 1, Edmonton, Canada, 2003.

[69] P. Koehn and J. Schroeder. Experiments in domain adaptation for statistical machine translation. In *ACL Workshop on Statistical Machine Translation*, 2007.

[70] S. Langer. Zur morphologie und semantik von nominalkomposita. In *Tagungsband der 4. Konferenz zur Verarbeitung naturlicher Sprache, KONVENS*, 1998.

[71] M. Larson, D. Willett, J. Kohler, and G. Rigoll. Compound splitting and lexical unit recombination for improved performance of a speech recognition system for german parliamentary speeches. In *6th International Conference on Spoken Language Processing (ICSLP)*, 2000.

[72] Y. Lee and N. Ge. Local reordering in statistical machine translation. In *Workshop of TCStar*, 2006.

[73] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics-Doklady 10, 707-710*, 1966.

[74] D. Lewis and J. Catlett. Heterogenous uncertainty sampling for supervised learning. In *ICML-94, pages: 148-156*, 1994.

[75] C. Li, M. Li, D. Zhang, M. Li, M. Zhou, and Y. Guan. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th ACL, pages: 720–727*, 2007.

[76] R. Liere and P. Tadepalli. Active learning with committees. In *Oregon State University*, Corvallis, OR, 2000.

[77] Y. Liu, Y. Huang, Q. Liu, and S. Lin. Forest-to-string statistical translation rules. In *Proceedings of ACL 2007, pages: 704-711*, Prague, Czech Republic, June, 2007.

[78] Y. Liu, Q. Liu, and S. Lin. Tree-to-string alignment template for statistical machine translation. In *Proceedings of COLING/ACL 2006, pages: 609–616*, Sydney, Australia, July, 2006.

[79] B. Maegaard, N. Bel, B. Dorr, E. Hovy, K. Knight, H. Iida, C. Boitet, and Y. Wilks. Machine translation. In *Hovy, E., Ide, N., Frederking, R., Mariani, J. Zampolli, A. (eds.), Multilingual Information Management: Current Levels and Future Abilities*, 1999.

[80] A. Mahmud and M. Khan. Building a foundation of hpsg-based treebank on bangla language. In *Proceedings of the 10th ICCIT*, Dhaka, December, 2007.

[81] D. Marcu, W. Wang, A. Echihabi, , and K. Knight. Spmt: Statistical machine translation with syntactified target language phrases. In *Proceedings of EMNLP-2006, pages: 44-52*, Sydney, Australia, 2006.

[82] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages: 133–139*, Philadelphia, PA, July, 2002.

[83] J. B. Mariño, R. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà. N-gram-based machine translation. In *Computational Linguistics, Volume 32, Number 4, pages: 527-549*, December 2006.

[84] A. K. McCallum. Mallet: A machine learning for language toolkit. In *http://mallet.cs.umass.edu*, 2007.

[85] D. I. Melamed. Statistical machine translation by parsing. In *Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics (ACL)*, Barcelona, Spain, 2003.

[86] C. Monz and M. de Rijke. Shallow morphological analysis in monolingual information retrieval for dutch, german, and italian. In *Second Workshop of the Cross-Language Evaluation Forum (CLEF)*, 2001.

[87] D. Munteanu and D. Marcu. Improving machine translation performance by exploiting comparable corpora. In *Computational Linguistics, 31 (4), pages: 477-504*, December, 2005.

[88] S. K. Naskar and S. Bandyopadhyay. A phrasal ebmt system for translating english to bangla. In *MT Summit X*, September 13-15, Phuket, Thailand, 2005.

[89] S. K. Naskar and S. Bandyopadhyay. Handling of prepositions in english to bengali machine translation. In *Third ACL-SIGSEM Workshop on Prepositions*, 2006.

[90] R. Nübel. Knowledge sources for the disambiguation of prepositions in machine translation. In *Proceedings of the Workshop on Future Issues for Multilingual Text Processing, Pacific Rim International Conference on Artificial Intelligence*, 1996.

[91] T. P. Nguyen and A. Shimazu. A syntactic transformation model for statistical machine translation. In *ICCPOL 2006: pages: 63-74*, 2006.

[92] S. Nießen and H. Ney. Morpho-syntactic analysis for reordering in statistical machine translation. In *Proceedings of MT Summit VIII, pages: 247-252*, Santiago de Compostela, Spain, September, 2001.

[93] S. Niessen and N. Ney. Statistical machine translation with scarce resources using morpho-syntactic information. In *Computational Linguistics, 30(2):181–204*, 2004.

[94] D. Oard and F. J. Och. Rapid-response machine translation for unexpected languages. In *MT SUMMIT IX, Proceedings of the Ninth Machine Translation Summit*, New Orleans, LO, September, 2003.

[95] F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the ACL, pages: 160–167*, 2003.

[96] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. A smorgasbord of features for statistical machine translation. In *Proceedings of HLT-NAACL*, 2004.

[97] F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *ACL 2002: Proc. of the 40th Annual Meeting of the Association for Computational Linguistics, pages: 295-302*, Philadelphia, PA, July, 2002.

[98] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics, pages: 311-318*, 2002.

[99] M. Popovic, D. Stein, and H. Ney. Statistical machine translation of german compound words. In *Proceedings of FinTAL - 5th International Conference on Natural Language Processing, pages: 616–624*, Turku, Finland, 2006.

[100] G. Pullum and R. Huddleston. Prepositions and prepositional phrases. In *In Huddleston and Pullum (eds.), pages: 597-661*, 2002.

[101] C. Quirk, A. Menezes, and C. Cherry. Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL), pages: 271–279*, Ann Arbor, Michigan, June, 2005.

[102] A. Rahman, S. Islam, A. Alim, and K. Hasan. A rule based english-bangla mt system for compound sentences. In *Proceedings of NCCPB*, 2005.

[103] G. Rashmi, R. D. Brown, and J. Carbonell. Active learning in example-based machine translation. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, NODALIDA 2009.

[104] G. Rauh. On the grammar of lexical and nonlexical prepositions in english. In *Zelinskiy-Wibbelt (eds.), pages: 99-150*, 1993.

[105] K. Rottmann and S. Vogel. Word reordering in statistical machine translation with a pos-based distortion model. In *TMI-2007: 11th International Conference on Theoretical and Methodological Issues in MT*, Skvde, Sweden, 2007.

[106] K. Rottmann and S. Vogel. Word reordering in statistical machine translation with a pos-based distortion model. In *TMI-2007: 11th International Conference on Theoretical and Methodological Issues in MT*, Skvde, Sweden, 2007.

[107] M. Roy. A semi-supervised approach to bengali-english phrase-based statistical machine translation. In *Proceedings of the Canadian Conference on AI, pages: 901–904*, 2009.

[108] M. Roy and F. Popowich. Phrase-based statistical machine translation for a low-density language pair. In *To Appear in Canadian Conference on AI*, 2010.

[109] M. Roy and F. Popowich. Word reordering approaches for bangla-english smt. In *To Appear in Canadian Conference on AI*, 2010.

[110] D. Samy, A. Moreno, and J. M. Guirao. A proposal for an arabic named entity tagger leveraging a parallel corpus. In *International Conference RANLP, pages: 459-465*, Borovets, Bulgaria, 2005.

[111] A. Sarkar, G. Haffari, and N. Ueffing. Transductive learning for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic. June 25-27, 2007.

[112] D. Shen, J. Zhang, J. Su, G. Zhou, and C. Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Barcelona, Spain, July 21-26, 2004.

[113] L. Shen, A. Sarkar, and F. J. Och. Discriminative reranking for machine translation. In *Proceedings of HLT-NAACL, pages: 177-184*, 2004.

[114] T. Sherif and G. Kondrak. Bootstrapping a stochastic transducer for arabic-english transliteration extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 864–871, Prague, Czech Republic, June 2007.

[115] T. Sherif and G. Kondrak. Substring-based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 944–951, Prague, Czech Republic, June 2007.

[116] H. Simpson, K. Maeda, and C. Cieri. Basic language resources for diverse asian languages: A streamlined approach for resource creation. In *Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP, pages: 55–62*, Suntec, Singapore, 2009.

[117] J. Sjöbergh and V. Kann. Finding the correct interpretation of swedish compounds a statistical approach. In *Proceedings of LREC-2004, pages: 899–902*, Lisbon, Portugal, 2004.

[118] J. M. Sopena, A. LLoberas, and J. L. Moliner. A connectionist approach to prepositional phrase attachment for real world texts. In *COLING-ACL, pages: 1233-1237*, 1998.

[119] R. Sproat, T. Tao, and C. Zhai. Named entity transliteration with comparable corpora. In *Proceedings of ACL*, 2006.

[120] B. Stalls and K. Knight. Translating names and technical terms in arabic text. In *Proceedings of the COLING-ACL Workshop on Computational Approaches to Semitic Languages*, 1998.

[121] A. Stolcke. Srilm—an extensible language modeling toolkit. In John H. L. Hansen and Bryan Pellom, editors, *Proceedings of the ICSLP, 2002*, volume 2, page 901–904. Denver, 2002.

[122] S. Stymne, M. Holmqvist, and L. Ahrenberg. Effects of morphological analysis in translation between german and english. In *Proceedings of the Third ACL Workshop on Statistical Machine Translation*, Columbus, Ohio, 2008.

[123] C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the 16th International Conference on Machine Learning, pages: 406–414*, Morgan Kaufmann, San Francisco, CA, 1999.

[124] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf. Accelerated dp based search for statistical translation. In *Fifth European Conference on Speech Communication and Technology, pages: 2667–2670*, Rhodos, Greece, September, 1997.

[125] C. Tillmann and T. Zhang. A block bigram prediction model for statistical machine translation. In *ACM Transactions Speech Language Processing, 4(3):6*, 2007.

[126] A. Trujillo. Locations in the machine translation of prepositional phrases. In *Proceedings of TMI-92, pages: 13-20*, 1992.

[127] N. Ueffing. Using monolingual source-language data to improve mt performance. In *Proceedings of the IWSLT*, 2006.

[128] N. Ueffing, M. Simard, S. Larkin, and J. H. Johnson. Nrc's portage system for wmt 2007. In *Proceedings of the ACL Workshop on SMT*, 2007.

[129] N. UzZaman, A. Zaheen, and M. Khan. A comprehensive roman (english) to bangla transliteration scheme. In *Proc. International Conference on Computer Processing on Bangla (ICCPB-2006)*, Dhaka, Bangladesh, 2006.

[130] C. Voss. Interlingua-based machine translation of spatial expressions. In *University of Maryland: Ph.D. Dissertation*, 2002.

[131] C. Wang, M. Collins, and P. Koehn. Chinese syntactic reordering for statistical machine translation. In *Proceedings of EMNLP-CoNLL, pages: 737–745*, 2007.

[132] D. Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. In *Computational Linguistics, 23(3):377-403*, 1997.

[133] F. Xia and M. McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistic*, Geneva, Switzerland, 2004.

[134] P. Xu, J. Kang, M. Ringgaard, and F. J. Och. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of NAACL-HLT'09, pages: 245–253*, Boulder, Colorado, June 2009.

[135] K. Yamada and K. Knight. A syntax-based statistical translation model. In *39th Annual Meeting of the Association for Computational Linguistics, pages: 523-530*, 2001.

[136] M. Yang and K. Kirchhoff. Phrase-based backoff models for machine translation of highly inflected languages. In *Proceedings of EACL*, 2006.

[137] A. S. Yeh and M. B. Vilain. Some properties of preposition and subordinate conjunction attachments. In *COLING-ACL, pages: 1436-1442*, 1998.

[138] R. Zens, F. J. Och, and H. Ney. Phrase-based statistical machine translation. In *KI - 2002: Advances in artificial intelligence, M. Jarke, J. Koehler, and G. Lakemeyer*, Eds. Springer Verlag, vol. LNAI, 18-32, 2002.

[139] Y. Zhang, R. Zens, and H. Ney. Improved chunk-level reordering for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT), pages: 21-28*, Trento, Italy, October 2007.