M.Sc. Engg. Thesis

Towards Achieving A Delicate Blending between Rule-based Translator and Neural Machine Translator for Bengali to English Translation

by

Md. Adnanul Islam (0416052015F)

Submitted to

Department of Computer Science and Engineering

(In partial fulfilment of the requirements for the degree of
Master of Science in Computer Science and Engineering)

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka 1000

November 6, 2019

Dedicated to my loving parents

Author's Contact

Md. Adnanul Islam
House-516, Road-2, Block-I,
Bashundhara R/A,
Dhaka
Email: `islamadnan2265@gmail.com`

The thesis titled "Towards Achieving A Delicate Blending between Rule-based Translator and Neural Machine Translator for Bengali to English Translation", submitted by Md. Adnanul Islam, Roll No. **0416052015F**, Session April 2016, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on November 6, 2019.

Board of Examiners

1. ________________________________

Dr. A. B. M. Alim Al Islam — Chairman (Supervisor)
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

2. ________________________________

Dr. Md. Mostofa Akbar — Member (Ex-Officio)
Head and Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

3. ________________________________

Dr. Mahmuda Naznin — Member
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

4. ________________________________

Dr. Muhammad Abdullah Adnan — Member
Assistant Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

5. ________________________________

Dr. Md. Mahbubur Rahman — Member (External)
Professor
Department of Computer Science and Engineering
Military Institute of Science and Technology, Dhaka.

The thesis titled "Towards Achieving A Delicate Blending between Rule-based Translator and Neural Machine Translator for Bengali to English Translation", submitted by Md. Adnanul Islam, Roll No. **0416052015F**, Session April 2016, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on November 6, 2019.

Board of Examiners

1. ________________________________
Dr. A. B. M. Alim Al Islam
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

Chairman
(Supervisor)

2. ________________________________
Dr. Md. Mostofa Akbar
Head and Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

Member
(Ex-Officio)

3. ________________________________
Dr. Mahmuda Naznin
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

Member

4. ________________________________
Dr. Muhammad Abdullah Adnan
Assistant Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

Member

5. ________________________________
Dr. Md. Mahbubur Rahman
Professor
Department of Computer Science and Engineering
Military Institute of Science and Technology, Dhaka.

Member
(External)

Candidate's Declaration

This is hereby declared that the work titled "Towards Achieving A Delicate Blending between Rule-based Translator and Neural Machine Translator for Bengali to English Translation", is the outcome of research carried out by me under the supervision of Dr. A. B. M. Alim Al Islam in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Adnan

Md. Adnanul Islam
Candidate

Acknowledgment

Foremost, I express my heart-felt gratitude to my supervisor, Dr. A. B. M. Alim Al Islam, for his constant supervision of this work. He helped me a lot in every aspect of this work and guided me with proper directions whenever I sought one. His patient hearing of my ideas, critical analysis of my observations, detecting flaws (and amending thereby) in my thinking, and writing have made this thesis a success.

I would also want to thank the respected members of my thesis committee: Dr. Md. Mostofa Akbar, Dr. Mahmuda Naznin, Dr. Muhammad Abdullah Adnan, and the external member Dr. Md. Mahbubur Rahman, for their encouragement, insightful comments, and valuable suggestions.

I am also thankful to Md. Saidul Hoque Anik (Lecturer, CSE, MIST). I sought help from him a number of occasions regarding simulation setup and performance evaluation of this thesis. Besides, I am grateful to Dr. Rifat Shahriyar (Associate Professor, CSE, BUET), Abhik Bhattacharjee, and Tahmid Hasan (Lecturer, CSE, BUET) for their kind support during my experimentation with a large dataset. In addition, I am grateful to Dr. Swakkhar Swatabta (Associate Professor, CSE, UIU) and Novia Nurain (Ph.D. student in CSE, BUET and Assistant Professor, CSE, UIU) for their help and valuable suggestions regarding the writing and presentation of this thesis.

Last but not the least, I remain ever grateful to my beloved parents, who always exist as sources of inspiration behind every success of mine.

Abstract

Although, a number of research studies have been done on natural language processing (NLP) in different areas such as Example-based Machine Translation (EBMT), Phrase-based Machine Translation, etc., for different pairs of languages such as English to Bengali translator, very few research studies have been done on Bengali to English translation. Popular and widely available translators such as Google translator performs reasonably well when translating among the popular languages such as English, French, or Spanish; however, they make elementary mistakes when translating the languages that are newly introduced to the system such as Bengali, Arabic, etc.

Google Translator uses Neural Machine Translation (NMT) approach with Recurrent Neural Network (RNN) to build its multilingual translation system. Prior to NMT, Google Translator used Statistical Machine Translation (SMT) approach. However, these approaches solely depend on the availability of a large parallel corpus of the translating language pairs. As a result, most of the research studies found so far on NLP have been performed keeping English as the base or source language. Here, a good number of widely spoken potential languages still remain nearly unexplored. Bengali, the eighth one in terms of usage all over the world, represents one of the prominent examples among those languages. Therefore, in this study, we explore improvized translation from Bengali to English. To do so, we study both the rule-based translator and the data-driven machine translators (NMT and SMT) in isolation, and in combination with different approaches of blending between them. More specifically, first, we implement some basic grammatical rules along with identification of names as subjects and optimization of Bengali verbs in our rule-based translator. Next, we integrate our rule-based translator with each of the data-driven machine translators (NMT and SMT) separately using different approaches. Besides, We perform rigorous experimentation over different datasets to reveal a comparison among the different approaches in terms of translation accuracy, time complexity, and space complexity.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human beings have been communicating using various spoken languages since their earliest days on the earth. Languages can express thoughts on an unlimited number of topics, e.g., social interaction, religion, past, future, etc. While many believe the number of languages in the world is about 6500, there are actually around 7106 living languages in the world [46]. Although this number might be the latest count, there is no clear answer on the exact number of languages that still exist. These huge number of languages are spread all over the world. For example, around 230 languages are spoken in Europe, whereas over 2000 languages are spoken in Asia [47].

Every human language has a vocabulary consisting of thousands of words, which are primarily built up from several dozens of speech sounds. More remarkable point here to be noted is that every normal child basically learns the whole system (mother tongue) just from hearing others using it. However, apart from the mother tongue, other languages are generally learnt in a more systematic process. Besides, in all languages, there are many words that may have multiple meanings and also some sentences may use different grammatical structures to express the same meaning [3]. This challenge, in turn, makes it immensely difficult to perform semantic analysis based translation between a pair of languages. Moreover, the task of translation experiences the top level of difficulty when the pair of languages contain a source language that is less explored in terms of having substantially large parallel corpus [1]. Bengali represents an example of such a source language. Therefore, it remains a great challenge to do the right semantic analysis to properly recognize any sentence of such a language.

To this context, in this thesis, we study Bengali to English machine translation by semantic based parts of speech tagging, verb identification and stemming, and name identification by lemmatization.

We perform our study through exploring rule-based translation and neural machine translation - both in isolation and in combination, through applying different blending approaches.

1.1 Motivation behind Our Work

Natural languages such as English, Spanish, and even Hindi are rapidly progressing in machine translation. While progress has been made in language translation software and allied technologies, the primary language of the ubiquitous and all-influential World Wide Web remains to be English [48]. English is mostly the language of latest applications, programs, new freeware, manuals, shareware, peer to peer conversation, social media networks, and websites [48].

Reading

Writing

Speaking and Listening

قوموں کی زندگی میں اتحاد و اتفاق ہی وہ طاقت ہے ، جس کی بدولت وہ مشکلات پر
قابو پا کر کامیاب ہوسکتی ہیں۔ قومی اتحاد ملک کی ترقی میں اہم کردار ادا کرتا ہے۔
قومی اتحاد ہماری قوم کو مضبوط بنانے کے لئے بہت اہم ہے۔ اس میں کوئی شک
نہیں ہےکہ پاکستان اپنی تاریخ کے سب سے نازک دور سے گزر رہا ہے۔ پاکستان
کوطاقتور اور خوشحال بنانے کے لئے قومی اتحاد ایک لازمی امر ہے۔ آپس کے
اختلافات ختم کرنے اور اتحاد و یگانگت کی جتنی ضرورت آج ہے، شاید پہلے کبھی
نہ تھی۔

In the life of the nations,national unity is the strength due to which a nation can succeed by overpowering the difficulities.National unity plays an important role in the development of the country.National unity is very important to strenghten our nation. There is no doubt that Pakistan is passing through the most critical period of its history.National unity is an essential fact to make Pakistan strong and prosperous.Today the need of eradication mutual contradictions and to have unity and harmony is more than ever.

Translating documents

Figure 1.1: Applications of translation

Millions of immigrants who travel the world from non-English-speaking countries every year, face the necessity of learning English to communicate in the language, since it is very important to enter and ultimately succeed in mainstream English speaking countries. The success gets comprehended when the learning covers all forms of reading, writing, speaking, and listening that eventually realize the process of translation encompassing a diversified set of applications (Figure 1.1).

Working knowledge of the English language can create many opportunities in international markets and regions. However, similar to many other non-English-speaking countries, a major group of Bengali speaking people from Bangladesh and India lacks proficiency in English [4]. This crisis is getting boosted over the period of time, as there is no well-developed translator till now for Bengali to English translation. Therefore, the importance of an efficient Bengali to English translator is noteworthy.

1.2 Approach of Our Study

We present an overview on the approach of our study in this section. Our initial focus is to explore building a rule-based translator. To do so, we implement some of the Bengali grammatical rules in our system along with detection of person, number, tense, etc., using semantic analysis. Besides, we accomplish optimization of Bengali verbs and identification of names as subjects to improve the performance of our rule-based translator. After that, we explore and implement the classical NMT (Neural Machine Translator). Then, we integrate rule-based translator and NMT using different approaches to investigate the best-possible translation performance. We measure the performance using three standard metrics of machine translation. Besides, we extend our experimentation (similar to NMT) by exploring another popular data-driven machine translation technology, Statistical Machine Translation (SMT), as it was used by popular Google Translator just before NMT [9]. Finally, we present the results of our experimentation both statistically and graphically, and also analyze them in details.

Next, we present an outline of our thesis. In Chapter 2, we highlight the related work in the field of natural language processing, specially on Bengali-English translation. Then, we briefly discuss classical neural machine translation approach in Chapter 2, which is a very important part of our study. Next, in Chapter 3, we present our implemented rule-based translator for Bengali to English translation. In addition to that, we propose our verb identification and memory optimization techniques in Chapter 3. Here, we implement modified Levenshtein's distance algorithm for root verb detection (Chapter 3.2.3). Finally, we investigate the performance of our proposed translator using three different blending approaches, which we discuss in Chapter 3.

In Chapter 4, we discuss the performance evaluation of our proposed mechanisms. Here, first, we show our experimental settings and different datasets. Next, we discuss our performance evaluation metrics (BLEU [19], METEOR [20], and TER [21]). Then, we present various experimental results (simulation outputs, graphs, tables, etc.) and our findings in Chapter 4. Furthermore, we perform a casual cross checking to our results with respect to human behaviour by conducting an online survey to identify which method(s) people generally follow (perhaps subconsciously) to translate from Bengali to English. Afterwards, we unfold our possible future studies in Chapter 6. In the end, we conclude by summarizing the problem and our contributions in this study in Chapter 7.

1.3 Our Contributions

Based on our work, our main contributions in this study are as follows:

- First, we propose a rule-based Bengali to English translator that implements some basic grammatical rules for Bengali to English translation. Our rule-based translator mainly focuses on Bengali grammar with some exceptional approaches such as finding standard form of verbs, identifying unknown words (names) as subjects, etc. Apart from processing simple sentences, our rule-based translator also considers basic complex and compound sentences. Besides, our rule-based translator identifies subjects with emphasizing tags properly to improve its overall translation performance.

- Afterwards, we integrate rule-based translator with existing NMT using different possible approaches. To do so, first, we implement the classical NMT. Designing a parallel corpus containing Bengali-English sentence pairs for training NMT is one of the toughest challenges that we face since Bengali is an extremely low resource language. Next, we blend our rule-based translator and NMT in three different ways to investigate the best-possible blending approach. Afterwards, similar to NMT, we implement SMT, and blend our rule-based translator with it to verify our best-possible blending approach.

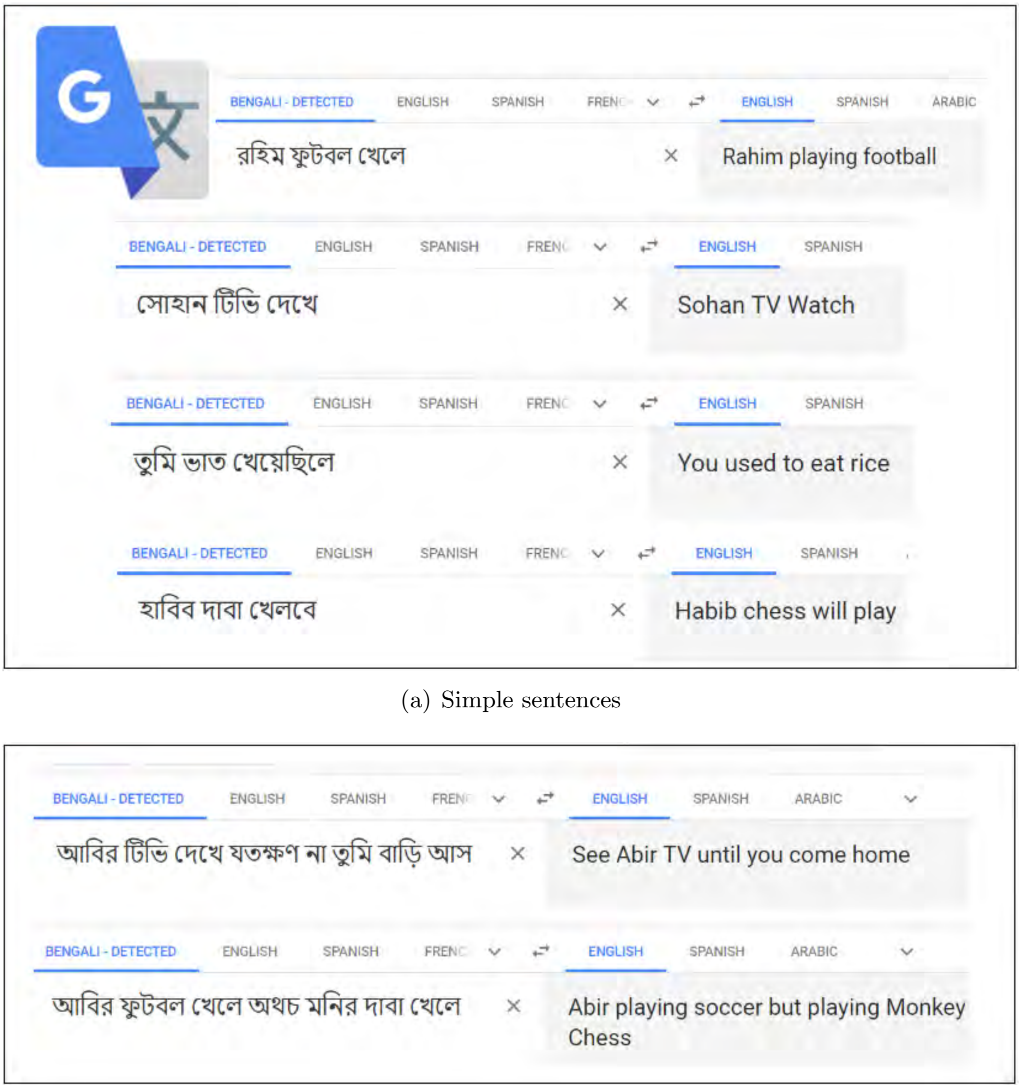
- Finally, we perform the performance evaluation for the proposed rule-based translator, classical NMT, and their integrated solutions using three standard metrics - BLEU, METEOR, and TER. We present the results for rule-based translator and NMT both in isolation and in combination. We also perform comparative analysis of the results among all the proposed approaches both statistically and graphically. Besides, we also show performance scores for SMT and its integrated solutions with rule-based translator as an extension of our experimental results.

Chapter 2

Background and Related Work

Bengali, being among the top ten languages worldwide, lags behind in some crucial areas of research in machine translation such as parts-of-speech (POS) tagging, text categorization and contextualization [35], syntax and semantic checking, etc. Most noteworthy previous studies in this regard include Example-based Machine Translation (EBMT) [4], phrase-based machine translation [5], syntactic transfer, and use of syntactic chunks as translation units [6]. However, these studies lack in processing Bengali words semantically. Besides, although significant research work can be found on English to Bengali translation [2][7], very few work has been performed on translating on the other way, i.e., from Bengali to English [3][8]. Popular translators such as Google, Bing, Yahoo Babel Fish, etc., often perform very poorly when they translate from Bengali to other languages. Google translator, the most popular one among them, uses neural machine translation (NMT) approach with RNN at present [9][10].

NMT has emerged as the most promising machine translation approach in recent years, showing superior performance on public benchmarks [1][11]. It is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional translation systems. In spite of the recent success of NMT in standard benchmarks, the lack of large parallel corpora poses a major practical problem for many language pairs such as Bengali-English [12]. This is why, NMT performs reasonably well when it translates among the most popular languages, however, it often makes elementary mistakes while translating languages that are less known to the system such as Bengali as shown in Figure 2.1 [14][15]. Focusing on rule-based translation in such a case might be a solution, which is yet to be explored in the literature. Moreover, blending NMT with such rule-based translator is yet another aspect to be investigated till now.



রহিম ফুটবল খেলে — Rahim playing football

সোহান টিভি দেখে — Sohan TV Watch

তুমি ভাত খেয়েছিলে — You used to eat rice

হাবিব দাবা খেলবে — Habib chess will play

(a) Simple sentences

আবির টিভি দেখে যতক্ষণ না তুমি বাড়ি আস — See Abir TV until you come home

আবির ফুটবল খেলে অথচ মনির দাবা খেলে — Abir playing soccer but playing Monkey Chess

(b) Complex sentence and compound sentence

Figure 2.1: Faulty translations of Google Translator

2.1 Existing Research Studies

Wu et al., [9] presented GNMT, Google's Neural Machine Translation system, with the objectives of reducing computational cost both in training and in translation inference, and increasing parallelism and robustness in translation. However, this approach solely relies on availability of significantly large parallel corpus and makes elementary mistakes while translating low-resource languages [36].

Artetxe et al., [11] removed the need of parallel data and proposes a novel method to train an NMT system with the objectives of relying on monolingual corpora only and profiting from small parallel corpora. Figure 2.2 reflects the architecture of this approach. Here, for each sentence in language L1, the system is trained alternating two steps: 1) denoising, which optimizes the probability of encoding a noised version of the sentence with the shared encoder and reconstructing it with the L1 decoder, and 2) on-the-fly back-translation, which translates the sentence in inference mode (encoding it with the shared encoder and decoding it with the L2 decoder), and then optimizes the probability of encoding this translated sentence with the shared encoder and recovering the original sentence with the L1 decoder. Training alternates between sentences in L1 and L2, with analogous steps for the latter. However, this promising approach still falls much behind the performance level of classical

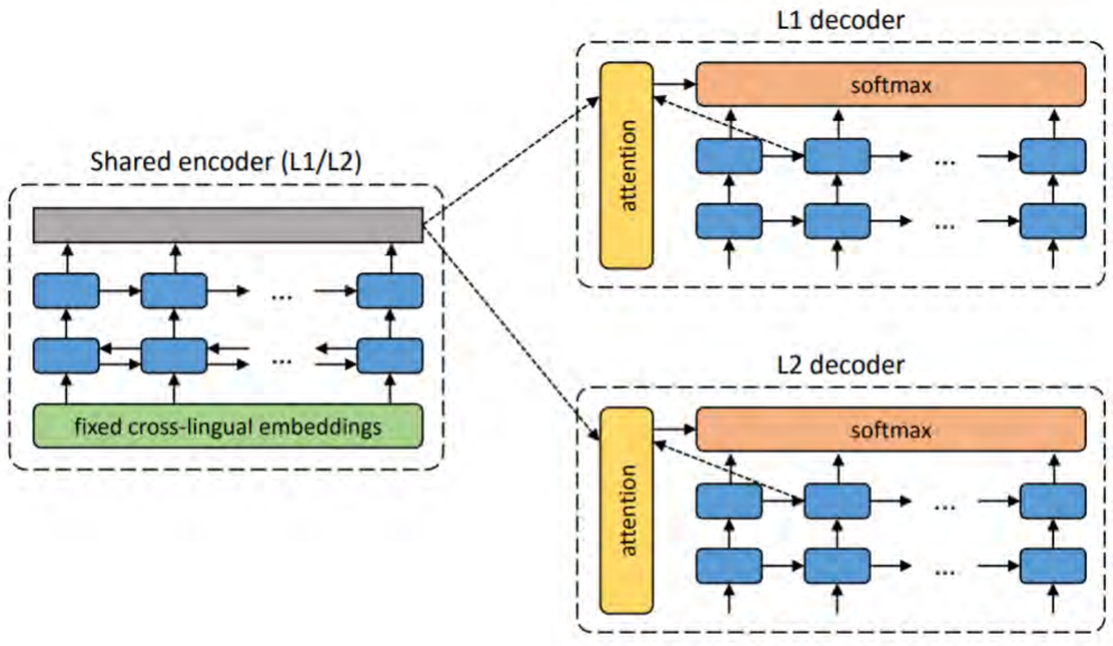


Figure 2.2: Architecture of Unsupervised Neural Machine Translation [11]

NMT. Gangadharaiah et al., [5] converted CNF to normal parse trees using bilingual dictionary with the objective of generating templates for aligning and extracting phrase-pairs for clustering. However, this work does not consider stemming of different forms of verbs and translation of unknown words. Besides, this approach also relies on availability of significantly large parallel corpus.

Saha et al., [28] reported an Example based Machine Translation System (EBMT) with the objective of translating news headlines from English to Bengali. However, this work was a specialized methodology only for newspaper headlines and did not consider the development of a Bengali lexicon with necessary tags. Kim et al., [6] used syntactic chunks as translation units with the objective of properly dealing with systematic translation for insertion or deletion of words between two distant languages. Figure 2.3 reflects the architecture of their proposed chunk-based EBMT. According to the architecture, first, given an input sentence, the system finds chunk (a sequence of words) sequence matches, and a chunk aligner finds their translations. Next, when no chunk match or no chunk alignment is found, it finds word/phrase matches, and uses a phrasal aligner to find the translations of them. Afterwards, it puts chunk translations and word/phrase translations into a lattice. Besides, for each translation, it keeps track whether the translation is from the chunk alignment or not. Finally, it performs standard beam decoding to find the best translation. However, this approach fails to address some basic grammatical rules during translation as it does not apply any specific rule during combining the chunk translations generated by the chunk aligner. Besides, this approach does not

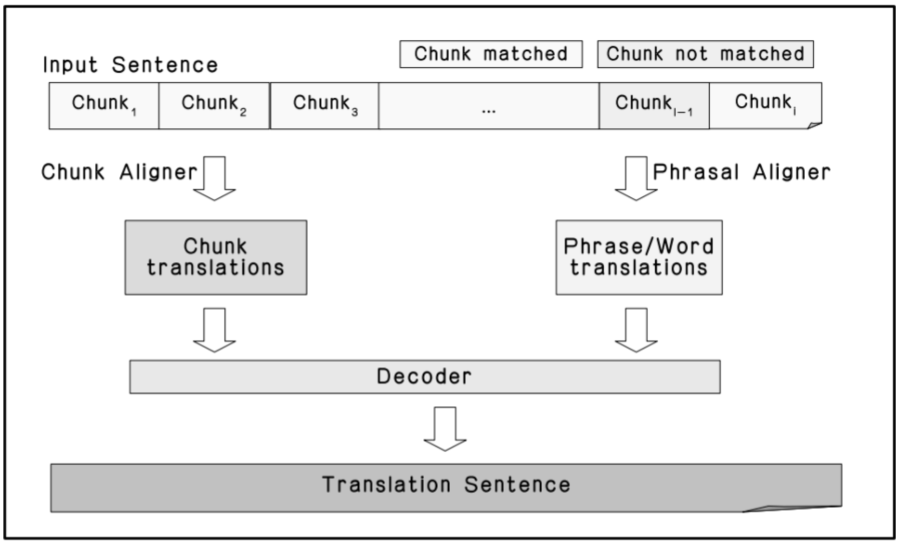


Figure 2.3: Architecture of chunk-based EBMT [6]

consider translating unknown words (not found in its vocabulary).

Additionally, there exist other research studies on NLP. For example, Souvik et al., [2] proposed a solution based on parse tree, Naskar et al., [31] handled prepositions in English, Dasgupta et al., [7] proposed another approach based on parse tree, etc. However, these techniques considered English-to-Bengali context only, not focusing on Bengali-to-English. Rahman et al., [3] explored statistical approach for Bengali-to-English translation. Besides, both Rahman et al., [8] and Alam et al., [23] explored a basic rule based approach for the same. However, these techniques either depended on a large corpus or omitted some basic grammatical features such as stemming and lemmatization. Apart from this limitation, these techniques are yet to consider an integration between rule-based translation and classical NMT.

Our work adopts implementation of GNMT as the classical NMT. Therefore, we discuss GNMT in the next section.

2.2 Google's Neural Machine Translation (GNMT) Model

Neural Machine Translation (NMT) is basically an end-to-end learning approach for automated translation. The strength of NMT lies in its ability to learn directly, in an end-to-end fashion, the mapping from input texts to associated output texts.

2.2.1 Background of GNMT

Google Translator, one of the most popular and widely available translators, earlier used Statistical Machine Translator (SMT) to build the multilingual translation system. SMT systems are not tailored to any specific pair of languages. In spite of being so promising and generalized, this approach usually does not work well for language pairs having significantly different word orders. Besides, SMT results may have superficial fluency that masks translation problems as SMT considers only a few words (chunk) from a source sentence at a time during translation [60]. Therefore, Google has moved towards NMT approach recently. Google's NMT model was first proposed by Wu et al., in 2016, which became a breakthrough in the field of NLP with the potential of addressing many shortcomings of traditional translation systems.

2.2.2 Architecture

Back in the old days, traditional phrase-based translation systems performed their task by breaking up source sentences into multiple chunks, and then translating them phrase-by-phrase. This led to less fluency or accuracy in the translation outputs and was not quite like how we, humans, perform the task of translation. We generally read the entire source sentence, understand its meaning, and then produce a translation. Neural Machine Translation (NMT) attempts to closely mimic that.

Specifically, an NMT system first reads the source sentence using an encoder to build a "thought" vector (a sequence of numbers that represents meaning of the sentence). Then, a decoder processes the sentence vector to perform a translation, as illustrated in Figure 2.4. This is often referred as the encoder-decoder architecture. In this manner, NMT addresses the local translation problem in the traditional phrase-based approach. Thus, NMT can capture long-range dependencies in languages, e.g., gender agreements, syntax structures, etc., and produce much more fluent translations as demonstrated by GNMT systems.

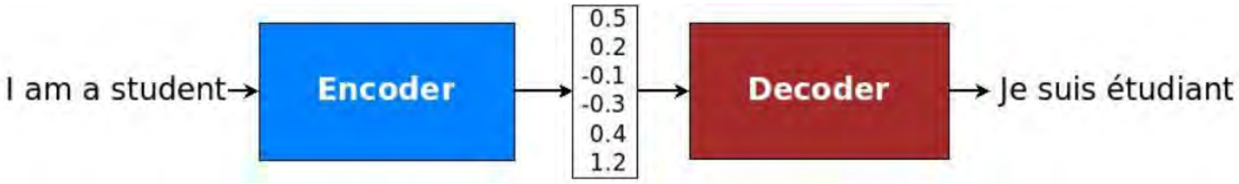


Figure 2.4: Encoder-decoder architecture for English to German translator [50]

NMT models vary in terms of their exact architectures. A natural choice for sequential data is the Recurrent Neural Network (RNN), used by most NMT models. Usually an RNN is used for both the encoder and the decoder. The RNN models, however, differ in terms of: (a) directionality – unidirectional or bidirectional [50]; (b) depth – single or multi-layer [50]; and (c) type – often either a vanilla RNN, a Long Short-term Memory (LSTM), or a gated recurrent unit (GRU) [50]. In our experimentation, a deep multi-layer RNN has been considered, which is unidirectional and uses LSTM as a recurrent unit. An example of such a model is shown in Figure 2.5.

In this example, a model is built to translate a source sentence "I am a student" into a target (German) sentence "Je suis étudiant". At a high level, the NMT model consists of two recurrent neural networks: the encoder RNN simply consumes the input source words without making any prediction; the decoder, on the other hand, processes the target sentence while predicting the next

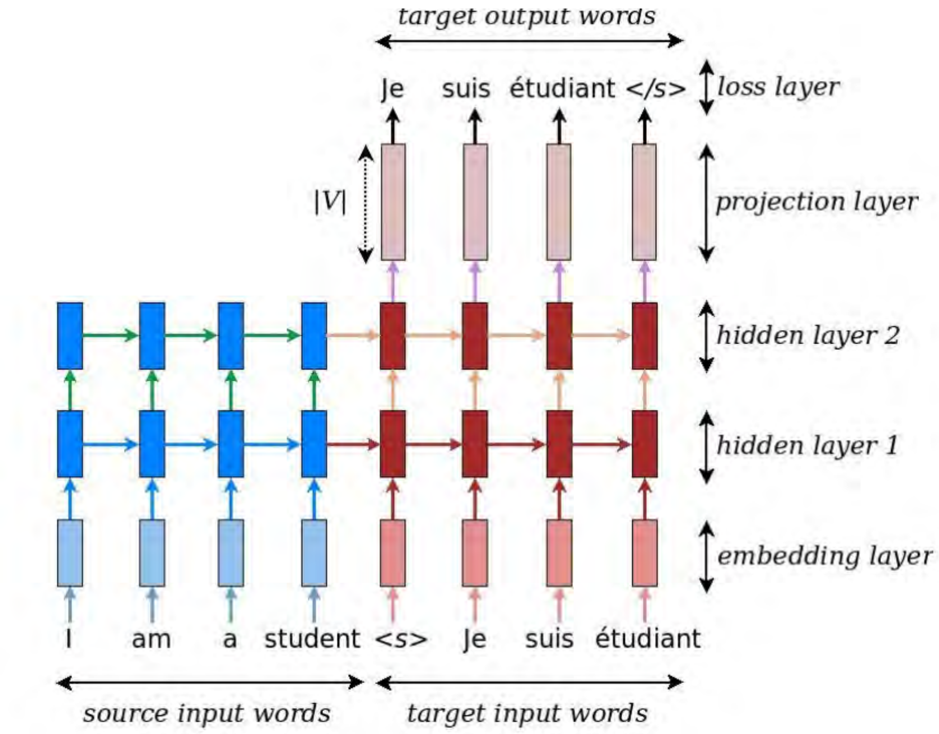


Figure 2.5: Neural machine translation – example of a deep recurrent architecture [50]

words. We present different components of the NMT's architecture below.

2.2.2.1 Embedding Layer

Initially, we need to train the NMT system using the bilingual parallel corpus. The model must first look up the source and target embeddings to retrieve the corresponding word representations. For this embedding layer to work, NMT chooses a vocabulary for each language first. Usually, NMT selects a vocabulary of size V, and treats only the most frequent V words as unique. It converts all other words to an "unknown" (<unk>) token and all get the same embedding. NMT learns the embedding weights, one set per language, during training with the parallel corpus.

2.2.2.2 Encoder

NMT can use one or more LSTM layers to implement the encoder model. The output of this model is a fixed-size vector that represents the internal representation of the input sequence. The number of memory cells in this layer defines the length of this fixed-sized vector.

Once retrieved, NMT then feeds the word embeddings as input into the main network, which consists of two multi-layer RNNs – an encoder for the source language and a decoder for the target language. These two RNNs, in principle, can share the same weights. However, in practice, the model often uses two different RNN parameters, which do a better job when fitting large training datasets. Here, the encoder RNN uses zero vectors as its starting states.

2.2.2.3 Decoder

The decoder must transform the learned internal representation of the input sequence into the correct output sequence. NMT can also use one or more LSTM layers to implement the decoder model. This model reads the fixed sized output generated by the encoder model. The decoder also needs to have access to the source information. Therefore, NMT simply initializes the decoder with the last hidden state of the encoder. Thus, as shown in Figure 2.5, NMT passes the hidden state of the last source word "student" to the decoder side.

2.2.2.4 Projection Layer

The Projection layer is a dense matrix to turn the top hidden states to logit[1] vectors of dimension having the vocabulary size. Projection layer maps the discrete word indices of an n-gram context to a continuous vector space as shown in Figure 2.6. NMT models share it such that for contexts containing the same word multiple times, the same set of weights apply to form each part of the projection vector.

2.2.2.5 Inference: Generating Translations

Once NMT has finished the training, it can generate translations from given previously unseen source sentences. This process is called inference. There is a clear distinction between training and inference (testing) since, at inference time, we only have access to the given source sentence. Afterwards, NMT performs the decoding.

The idea is simple as illustrated in Figure 2.6. Firstly, NMT encodes the source sentence in the same way as done during training. Next, it starts decoding as soon as it receives the starting symbol (<s>). Then, for each timestep on the decoder side, NMT treats the RNN's output as a set of logits.

[1] Logits generally refers to the unnormalized final scores of a machine learning model. We apply softmax to it to get a probability distribution over the classes. In our model, logits refers to the scores of words in vocabulary to appear in the translation.

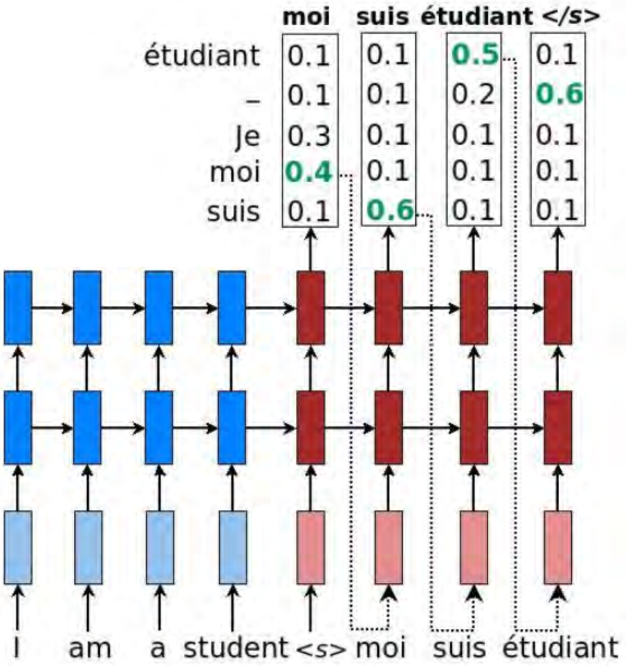


Figure 2.6: Neural machine translation – inference [50]

NMT then chooses the most likely word, the id associated with the maximum logit value, as the emitted word. For example, in Figure 2.6, the word "moi" has the highest translation probability in the first decoding step. Afterwards, NMT feeds this word as an input to the next timestep. This step is what makes inference different from training. Finally, the process continues until the decoder produces the end-of-sentence marker (</s>) as an output symbol.

GNMT system approaches the accuracy achieved by average bilingual human translators on some of the designed test sets. In particular, compared to the previous phrase-based production system, this GNMT system delivers roughly a 60% reduction in translation errors on several popular language pairs [50]. However, even such a promising approach exhibits some major weaknesses. Three inherent weaknesses of NMT are: 1) its slower training and inference speed, 2) ineffectiveness in dealing with rare words, and 3) sometimes failure to translate all words in the source sentence. Its performance generally improves with the increased size of dataset (parallel corpus). We investigate the integration of NMT with rule-based approach to improve the overall performance in the next chapters.

2.3 Limitations of Existing Research Studies

Although there exists a significant number of research in the field of language processing, Bengali remains little explored in the literature. Therefore, in this study, we specifically address some major limitations in Bengali to English translation along with Bengali language processing to some extent. These major limitations are discussed as follows.

- None of the existing studies focuses on integration of rule-based translator with any data-driven machine translator (NMT or SMT) for translation between any language pair. Investigation of possible outcomes of such integration or blending between these two translation approaches is completely absent in the literature. However, our survey (in Chapter 5) suggests that most people generally prefer using both NMT (or SMT)-like and grammatical rules based translations while translating from one language to another in their daily life. This human behaviour points towards a prospect of exploring integration between rule-based translator and NMT in machine translation.

- Apart from this, a large parallel corpus for Bengali-to-English machine translation is yet to be available. However, the performance of NMT solely relies on availability of significant amount of training data. The more example translation NMT sees (during training), the better it translates (infers). However, not a single corpus containing a substantial number of Bengali-English sentence pairs is available for NMT, which significantly limits the performance of NMT in translating Bengali to English.

- Besides, existing techniques do not consider finding stems of different forms of Bengali verbs. As Bengali verbs can take multiple forms based on tenses, we need to detect the standard form of verb by stemming for optimizing total memory consumption. Current literature does not consider this issue not only for Bengali but also for other languages.

- In addition to that, existing studies cannot properly recognize and translate words, which we cannot find in specialized vocabulary such as names of people. Besides, Bengali sentences may contain emphasizing tags attached to the subjects as suffixes, which leads to faulty detection of subjects as names. Existing studies have not explored this issue too.

Chapter 3

Proposed Methodology

Our work initially focuses on building a rule-based translator [15]. Next, our target is to explore and implement the classical NMT, i.e., GNMT. To do so, we collect and build datasets (Bengali and English language pair) of different sizes from different sources. Subsequently, after implementing both rule-based translator and classical NMT in isolation, we integrate these two translators using different approaches to investigate the best possible translation performance. We present our proposed mechanisms and algorithms next in details.

3.1 Rule-based Translator

Our rule-based translator initially focuses on implementation of simple sentences. Here, simple sentence analysis and recognition is the preliminary step which leads to the advancement of our system towards implementation of complex and compound sentences later. However, analyzing and recognizing a simple sentence of a language requires enormous knowledge on that particular language. In this study, our rule-based translator particularly implements some basic grammatical rules for Bengali to English translation. Figure 3.1 illustrates the mechanism of our proposed rule-based translator. As shown in the figure, our proposed mechanism for implementing the rule-based translator basically consists of six major steps. We elaborate each of the steps in the following subsections.

3.1.1 Step-1: Input of Bengali Text

The first step is to take an input sentence. Here, the input sentence is a Bengali sentence. If we get a paragraph as input, we recognize sentences by splitting the input paragraph through the sentence

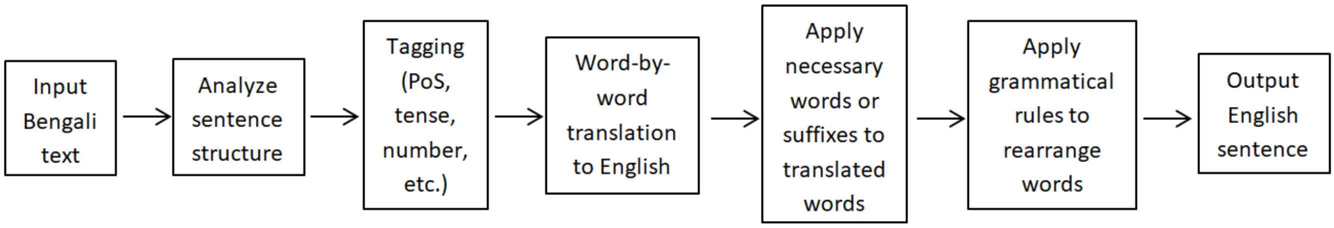

Input Bengali text
Analyze sentence structure
Tagging (PoS, tense, number, etc.)
Word-by-word translation to English
Apply necessary words or suffixes to translated words
Apply grammatical rules to rearrange words
Output English sentence

Figure 3.1: Mechanism of our proposed rule-based translator

terminating delimiters. Our considered sentence terminating delimiters are - '|' and ';'. The input sentence is, then, fed to the tokenizer for token identification and further processing. Figure 3.2 shows an example of how a Bengali paragraph can appear as input.

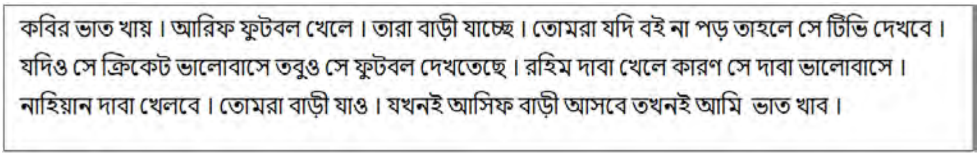

কবির ভাত খায় । আরিফ ফুটবল খেলে । তারা বাড়ী যাচ্ছে । তোমরা যদি বই না পড় তাহলে সে টিভি দেখবে ।
যদিও সে ক্রিকেট ভালোবাসে তবুও সে ফুটবল দেখতেছে । রহিম দাবা খেলে কারণ সে দাবা ভালোবাসে ।
নাহিয়ান দাবা খেলবে । তোমরা বাড়ী যাও । যখনই আসিফ বাড়ী আসবে তখনই আমি ভাত খাব ।

Figure 3.2: Input paragraph

We split the paragraph into several independent sentences. For example, the paragraph in Figure 3.2 gets split into nine sentences as shown in Figure 3.3. We consider these sentences as separate input sentences (that we need) to translate one by one. Therefore, we tokenize each sentence next.

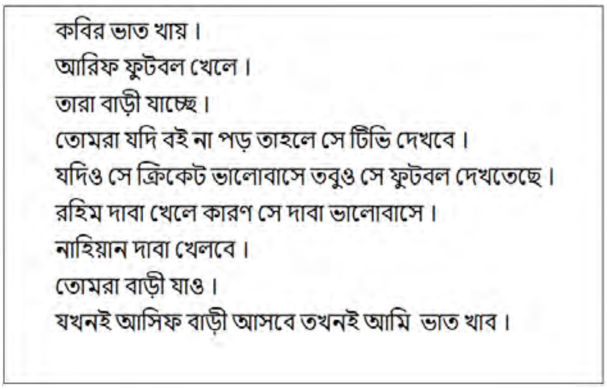
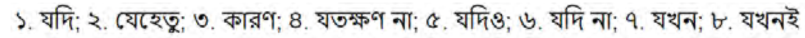

কবির ভাত খায় ।
আরিফ ফুটবল খেলে ।
তারা বাড়ী যাচ্ছে ।
তোমরা যদি বই না পড় তাহলে সে টিভি দেখবে ।
যদিও সে ক্রিকেট ভালোবাসে তবুও সে ফুটবল দেখতেছে ।
রহিম দাবা খেলে কারণ সে দাবা ভালোবাসে ।
নাহিয়ান দাবা খেলবে ।
তোমরা বাড়ী যাও ।
যখনই আসিফ বাড়ী আসবে তখনই আমি ভাত খাব ।

Figure 3.3: Input sentences obtained from the paragraph shown in Figure 3.2

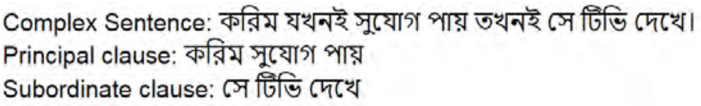
In our rule-based system, we cover mostly simple sentences along with basic complex and compound sentences. To differentiate simple sentences from complex and compound sentences, we primarily check whether any of the keywords of complex or compound sentence is present in the input sentence. The rationale behind this consideration is the fact that a complex sentence is formed when we join a principal clause and a subordinate clause with a connective. It can have one or more dependent clauses (also called subordinate clauses). Since a dependent clause cannot stand on its own as a sentence, complex sentences must also have at least one independent clause. Therefore, a complex sentence is basically a union of two simple sentences that come out of the clauses. Our system can recognize the following popular basic keywords of a complex sentence as shown in Figure 3.4.



১. যদি; ২. যেহেতু; ৩. কারণ; ৪. যতক্ষণ না; ৫. যদিও; ৬. যদি না; ৭. যখন; ৮. যখনই

Figure 3.4: Keywords of complex sentence under our considerations

If our system determines a complex sentence by matching any of these keywords then it splits the sentence into two grammatical clauses - 1) Principal clause, and 2) Subordinate clause. Each clause then acts as an independent simple sentence, which then passes to the tokenizer in the next phase for further processing. For example, we split a complex sentence into the clauses as shown in Figure 3.5.



Complex Sentence: করিম যখনই সুযোগ পায় তখনই সে টিভি দেখে।
Principal clause: করিম সুযোগ পায়
Subordinate clause: সে টিভি দেখে

Figure 3.5: Splitting a complex sentence into a principal clause and a subordinate clause

3.1.2 Step-2: Analysis of Sentence Structure and Tokenization

Next, we tokenize the sentences identified or splitted in the previous section. Our system, here, considers each Bengali word of a sentence as a token, and tags the tokens in various ways such as position, person, number, Parts of Speech (PoS), and tense. For example, let us consider the following input Bengali sentence:

সে ভাত খায়।

Our system will tokenize this input sentence as follows:

১. সে ; ২. ভাত ; ৩. খায় ; ৪. ।

Besides, our system initially determines the position of a token by analyzing a very basic grammatical rule for Bengali sentence formation - “Subject + Object + Verb”. We present it in Figure 3.6.

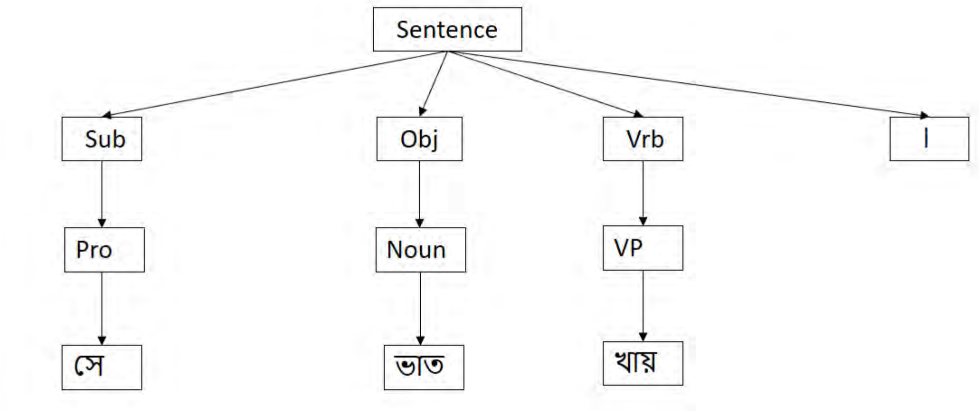


Figure 3.6: Bengali parse tree

The figure illustrates a Bengali parse tree that presents how our system recognizes the tokens with their roles in the sentence. Here, we have some predefined commonly-used nouns, pronouns, verbs, etc., in our system that are presented in the parse tree as “Noun”, “Pro”, “VP”, etc., respectively. Our system accomplishes this task by its token tagging process, which we discuss next.

3.1.3 Step-3: Token Tagging

Our system performs the task of token tagging using the information from grammatical set of rules for Bengali sentences. In our token tagging, we identify position, PoS, number, and person for each of the tokens. Table 3.1 illustrates an example of initial token tagging for our previous input sentence.

Token	Position	POS	Number	Person
সে	Sub	PRO	1	3
ভাত	Obj	Noun	Null	Null
খায়	Vrb	VP	Null	Null
।	Delim	Null	Null	Null

Table 3.1: Initial tagging table for tokens

3.1.4 Step-4: Word-by-word Translation

Our rule-based translation system consists of a vocabulary containing around 1,000 commonly-used Bengali-English word pairs. Our system performs direct translation using this vocabulary and necessary information extracted from the token tagging. Table 3.2 shows how our system performs word by word translation for each of the tokens.

Token	Translation
সে	He
ভাত	Rice
খায়	Eat
।	.

Table 3.2: Token translation using vocabulary

This results in our updated and final token tagging with translation as shown in Table 3.3.

Token	Position	POS	Number	Person	Translation
সে	Sub	PRO	1	3	He
ভাত	Obj	Noun	Null	Null	Rice
খায়	Vrb	VP	Null	Null	Eat
।	Delim	Null	Null	Null	.

Table 3.3: Final tagging with translation containing necessary information about each token

3.1.5 Step-5: Apply Necessary Words and Suffixes

Next, our system can determine the tense of an input sentence by analyzing the suffixes of Bengali verbs. We do this through maintaining a list of commonly-used Bengali suffixes mapped to a particular tense or tense code. Table 3.4 shows only a partial scenario of how we map the suffixes to different tenses. At this stage, we need to deal with Bengali verbs having multiple forms since one standard Bengali verb can take multiple forms based on its tenses in different sentences. Therefore, optimization of vocabulary for verbs becomes an issue in terms of total memory consumption by the system. We will address this issue later in details in Chapter 3.2. Here, we present an example in Figure 3.7 where a Bengali verb has been processed to determine the tense by removing its suffix.

Bengali Suffix	Tense Code	Tense
ই	11	Simple Present
চ্ছে	12	Present Continuous
য়াছে	13	Present Perfect
য়াছি	13	Present Perfect
ছিলে	21	Simple Past
ছিলাম	21	Simple Past
চ্ছিল	22	Past Continuous
বে	31	Simple Future
ব	31	Simple Future

Table 3.4: Commonly-used Bengali suffixes representing tenses

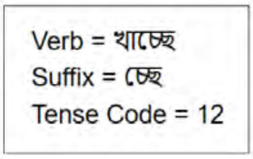

Verb = খাচ্ছে
Suffix = চ্ছে
Tense Code = 12

Figure 3.7: Determining tense from a Bengali verb

After we detect the tense, our system modifies the translated English verb by adding necessary suffixes and words (auxiliary verbs) using information extracted from the token tagging such as number and person of subject.

Table 3.5 shows how our system modifies the translated verbs depending on tense, number, and person of subject obtained from the token tagging table.

Tense	**Tense Code**	**Person**	**Number**	**Verb Modification** (adding words and suffixes)
Simple Present	11	1	1/ 2	Null
		2	1/ 2	Null
		3	1	Add ‘es/ s’
		3	2	Null
Present Continuous	12	1	1	am + ‘ing’ form
		1	2	are + ‘ing’ form
		2	1/ 2	are + ‘ing’ form
		3	1	is + ‘ing’ form
		3	2	are + ‘ing’ form
Present Perfect	13	1	1/ 2	have + ‘past participle’ form
		2	1/ 2	have + ‘past participle’ form
		3	1	has + ‘past participle’ form
		3	2	have + ‘past participle’ form
Simple Past	21	1/ 2/ 3	1/ 2	‘past’ form
Past Continuous	22	1/ 2/ 3	1	was + ‘ing’ form
		1/ 2/ 3	2	were + ‘ing’ form
Past Perfect	23	1/ 2/ 3	1/ 2	had + ‘past participle’ form
Simple Future	31	1	1/ 2	shall + verb
		2/ 3	1/ 2	will + verb
Future Continuous	32	1	1/ 2	shall be + ‘ing’ form
		2/ 3	1/ 2	will be + ‘ing’ form
Future Perfect	33	1	1/ 2	shall have + ‘past participle’ form
		2/ 3	1/ 2	will have + ‘past participle’ form

Table 3.5: Modifying verbs based on tenses, persons, and numbers

3.1.6 Step-6: Rearrange Words by Applying Grammatical Rules

Our system has generated all the words of the translated sentence by now. However, we need to arrange these words according to grammatical rules of the target language, i.e., English. More specifically, we need to apply grammatical rules for building an English sentence as we are translating into English. Basic rule for building a simple sentence in English is - “Subject + Verb + Object”. Therefore, our system arranges the translated words accordingly using the token tagging table. Figure 3.8 illustrates how the target sentence gets an ordered list of translated words from the input sentence. Here, the previous input sentence has been taken as an example for generating the translation.

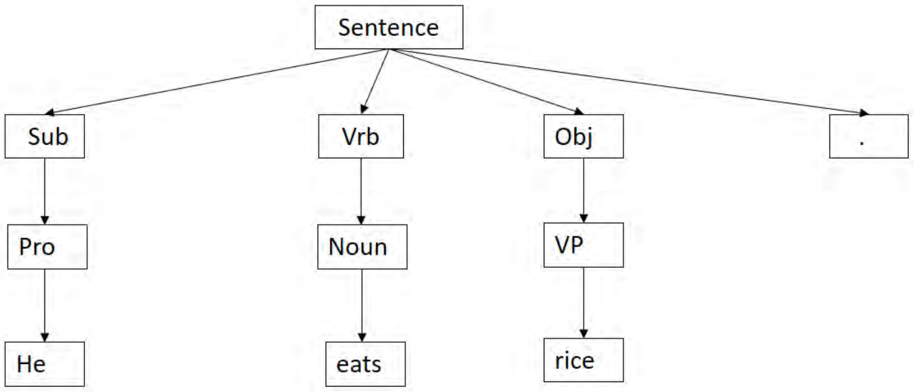
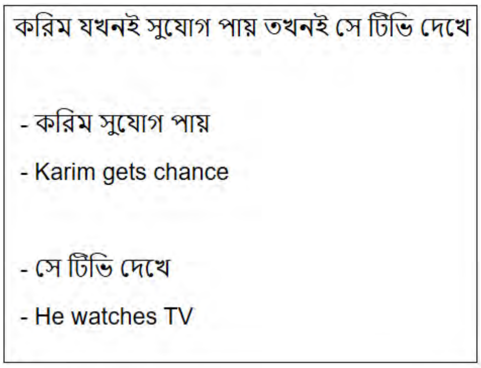


Figure 3.8: Target (English) parse tree

For complex and compound sentences, our system generates two translated sentences for two different clauses (similar to simple sentences) separately as discussed earlier. The following sentence in Figure 3.9 is an example to show how our system processes a complex sentence by splitting it into two simple sentences first. Afterwards, our system adds necessary English merging keywords for corresponding Bengali merging keywords at right places so that these two simple sentences merge to form the target complex or compound sentence as shown in Figure 3.10 and Figure 3.11 respectively.

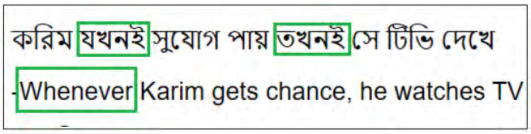


করিম যখনই সুযোগ পায় তখনই সে টিভি দেখে

- করিম সুযোগ পায়
- Karim gets chance

- সে টিভি দেখে
- He watches TV

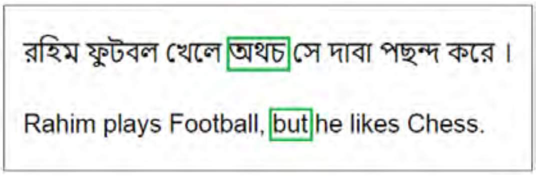
Figure 3.9: Processing a complex sentence into two clauses representing two simple sentences



করিম যখনই সুযোগ পায় তখনই সে টিভি দেখে

Whenever Karim gets chance, he watches TV

Figure 3.10: Translation of a complex sentence



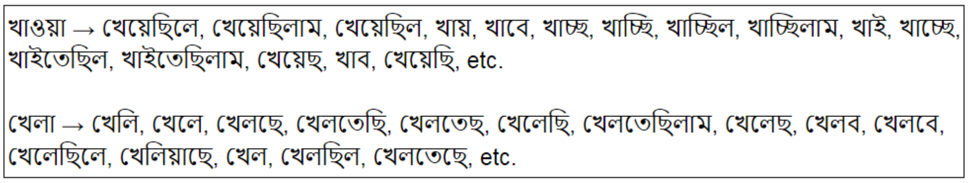
রহিম ফুটবল খেলে অথচ সে দাবা পছন্দ করে।

Rahim plays Football, but he likes Chess.

Figure 3.11: Translation of a compound sentence

3.2 Verb Identification and Memory Optimization

In our proposed rule-based translation system, we store the verbs in a database along with the other words as a part of the vocabulary of the intended (Bengali) language. It is worth mentioning that, in Bengali, one verb may have multiple representations based on tense and subject of a sentence as shown in Figure 3.12. The figure shows an example of different forms taken by each of the two different verbs in Bengali, which correspond to ‘eat’ and ‘play’ in English respectively.



খাওয়া → খেয়েছিলে, খেয়েছিলাম, খেয়েছিল, খায়, খাবে, খাচ্ছ, খাচ্ছি, খাচ্ছিল, খাচ্ছিলাম, খাই, খাচ্ছে, খাইতেছিল, খাইতেছিলাম, খেয়েছ, খাব, খেয়েছি, etc.

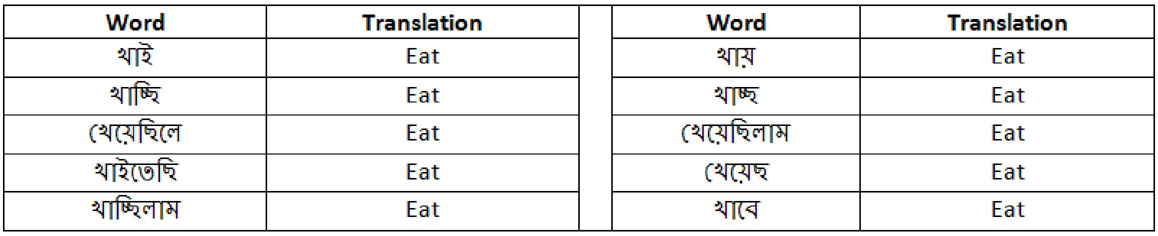
খেলা → খেলি, খেলে, খেলছে, খেলতেছি, খেলতেছ, খেলেছি, খেলতেছিলাম, খেলেছ, খেলব, খেলবে, খেলেছিলে, খেলিয়াছে, খেল, খেলছিল, খেলতেছে, etc.

Figure 3.12: Several forms of two different verbs in Bengali

We, therefore, explore three different approaches for translating such verbs efficiently in terms of memory consumption and accuracy [13]. We will discuss these approaches in the subsequent sections. Here, one approach improves over another in a sequential manner as we progress onward.

3.2.1 Approach 1: Plain Vocabulary including All Forms of Verbs

The first one is the simplest and most straightforward proposed approach in our implementation as presented in [14]. Here, similar to all other general words (nouns, pronouns, etc.), we simply insert all different forms of each standard verb with their standard translation as separate entries in the database (vocabulary). Table 3.6 illustrates how several entries for a standard verb are incorporated in the database or vocabulary.



Word	Translation		Word	Translation
খাই	Eat		খায়	Eat
খাচ্ছি	Eat		খাচ্ছ	Eat
খেয়েছিলে	Eat		খেয়েছিলাম	Eat
খাইতেছি	Eat		খেয়েছ	Eat
খাচ্ছিলাম	Eat		খাবে	Eat

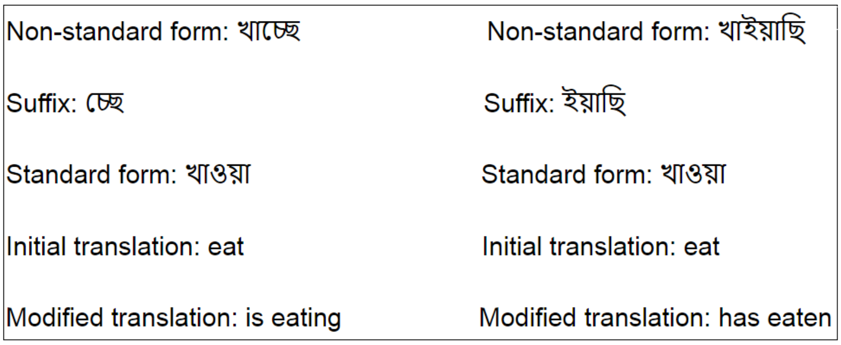
Table 3.6: Database table for translating a verb having different forms

Using this table, we can find the standard translated verb ('eat' in this case) for all different forms of the verb, which we then modify according to the tense and subject of the sentence applying semantic analysis as discussed earlier. Let us consider an example of a translated verb 'eat'. We will process the verb as 'is eating', 'ate', 'has eaten', etc., based on the semantic analysis (using token tagging table) of the sentence. This approach guarantees 100% accuracy in terms of verbs translation. However, memory consumption becomes a major issue due to the repetitive insertions of one standard verb in various forms resulting in wastage of considerable chunk of space.

3.2.2 Approach 2: Optimized Database with Semantic Analysis

Our next proposed approach for verb identification offers an immediate improvement over our previous approach as presented in [14]. As discussed earlier, if it is required to store the word translation for each form of the same verb then the database will become very large due to the repetitive insertions leading to massive unnecessary memory consumption. However, we can avoid such multiple insertions of the same verb (having different forms) in our database through an optimization technique with semantic analysis approach.

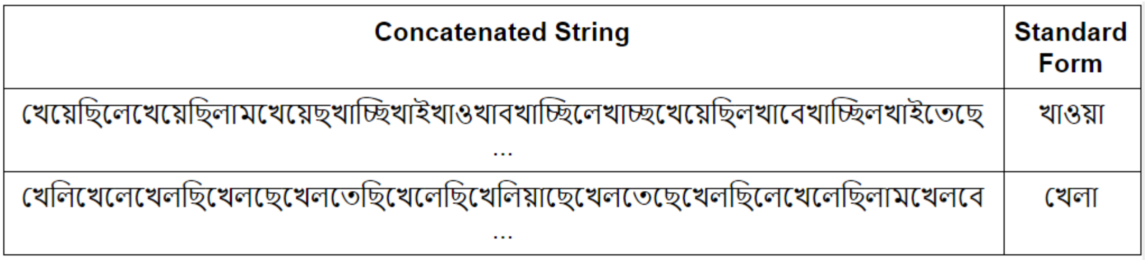
Here, we store only the standard verb in the vocabulary. Afterwards, we apply semantic analysis to detect the standard form from the other forms of the verb depending on number, person, and tense as shown in Figure 3.13. The figure shows how one word (standard verb) can take two different forms and suggests insertion of only corresponding particular standard word in the database omitting the need of inserting all of its different forms. This approach, thus, avoids multiple insertions in the database for the same verb with multiple forms.



Non-standard form: খাচ্ছে	Non-standard form: খাইয়াছি
Suffix: চ্ছে	Suffix: ইয়াছি
Standard form: খাওয়া	Standard form: খাওয়া
Initial translation: eat	Initial translation: eat
Modified translation: is eating	Modified translation: has eaten

Figure 3.13: Database optimization using semantic analysis on different forms of a verb

However, to detect that standard verb from its other different forms, we concatenate all the different forms of the verb as a single large string and insert it into another table with its standard form as a single entry. Figure 3.14 shows a couple of examples of such strings with corresponding standard form.



Concatenated String	Standard Form
খেয়েছিলেখেয়েছিলামখেয়েছখাচ্ছিখাইখাওখাবখাচ্ছিলেখাচ্ছখেয়েছিলখাবেখাচ্ছিলখাইতেছে ...	খাওয়া
খেলিখেলেখেলছিখেলছেখেলতেছিখেলেছিখেলিয়াছেখেলতেছেখেলছিলেখেলেছিলামখেলবে ...	খেলা

Figure 3.14: Mapping between concatenated string and corresponding standard form of two different verbs

This approach significantly improves over the previous approach in terms of searching time. Besides, this approach avoids overheads for multiple entries for one standard verb. Accuracy of detecting the verb still remains at the maximum (100%) in this approach too. However, it offers no significant improvement in terms of overall memory consumption since it ultimately stores all the forms (as a single string) of a standard verb in the database.

3.2.3 Approach 3: Modified Levenshtein Distance

Significant improvement is achieved with our final approach for verb identification in terms of both memory consumption and computational time as presented in [13]. In this approach, the translation of a verb is performed using a hash table. The key-value pair in the hash table consists of only the standard form of verbs of both source and target languages. In order to translate effectively, it is required to recognize these standard forms of verbs from their non-standard forms. For this purpose, a modified version of a popular string similarity measurement algorithm [17] is used in this approach, which is known as Levenshtein Distance [18].

As presented in [13], a non-standard form of a Bengali verb may have prefix and suffix assimilated into it based on tense, number, person, etc. Accordingly, instead of directly trying to match a non-standard form of verb with its standard form, first, the non-standard form is broken down into its root word (stemming) in this approach. Afterwards, that root word (stem) is matched with its standard form. Finally, translation of that non-standard form of verb can be obtained from its standard form. Detail of the whole approach can be found in [13].

3.3 Name Identification

One major and unique improvement achieved by our rule-based translation system is dealing with unknown words (not found in vocabulary), specifically name identification and corresponding translation technique [14]. Here, we need to identify the names of persons or objects to properly analyze the tokens obtained from an input sentence. Names of people are generally considered as nouns in the sentences, which dictate person, number, and gender of the subject if they appear as the subjects of the sentences. Therefore, properly identifying the names as subjects is very crucial for accurate translation. Google translator sometimes completely misunderstands the Bengali names of persons. As a consequence, it fails to recognize number and person of the subject, which ultimately leads to failure in translating even very basic Bengali sentences as shown in Figure 2.1 earlier. Besides, it is not practical to translate names by using any database containing the vocabulary.

In our proposed model, our system first recognizes the names by applying its specific grammatical rule set to identify the subjects not found in vocabulary. We show an example of this procedure of name identification while translating from Bengali to English in Figure 3.15. When our system detects the names as subjects in this way, it recognizes the token as a subject with tags - third person and singular number. Then our system can modify the verbs by adding prefixes and/or suffixes accordingly as discussed earlier. However, we are left with no translation for the names as names cannot appear in vocabulary. Therefore, we develop a Bengali to English phonetic mapping conversion system in our proposed system, which enables translation of the names (unknown words) from Bengali to English.

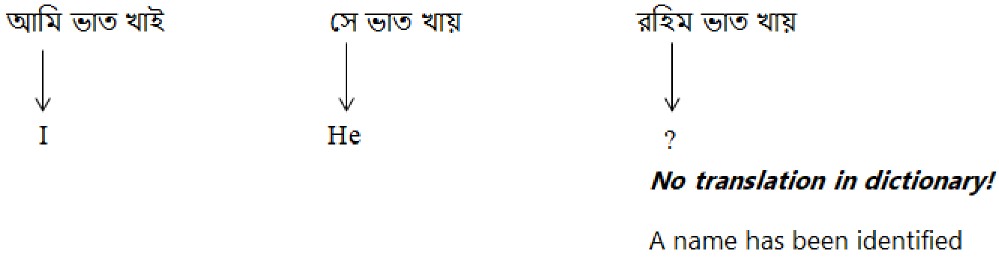


Figure 3.15: Name identification in our proposed model

Here, first, our system performs direct character-to-character(s) translation using a predefined set of character-to-character(s) mappings between two languages as shown in Figure 3.16.

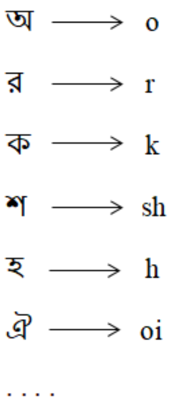


Figure 3.16: Character-to-character(s) mappings

Next, our system modifies the previously generated translation by introducing some missing characters (if any). To do so, our system checks whether two consonants appear in the translation consecutively. If our conversion system detects any such case, it inserts a vowel - 'a' or 'o' between those two consonants since this is how we generally translate Bengali names to English names. Figure 3.17 presents two example Bengali names translated using our proposed name translation technique. This proposed system cannot work in case of having emphasizing tags, which needs a specialized treatment as presented in the next subsection.

3.3.1 Subjects with Emphasizing Tags

There are different emphasizing tags used in many languages. The emphasizing tags, associated with the names or pronouns, are not actually any part of the main word (name or pronoun). Rather they

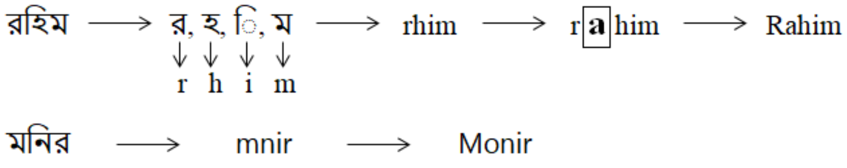


Figure 3.17: Translating names from Bengali to English using our proposed phonetic mapping conversion system

mean to emphasize the names or pronouns presenting a notion of supporting adjective or adverb. Thus, the tags have separate meanings and use in the sentences. If we do not identify these tags correctly and separate them from the names or pronouns, the resulting translation may become faulty as shown in Figure 3.18. In this figure, the system misinterprets the subject of the first sentence as

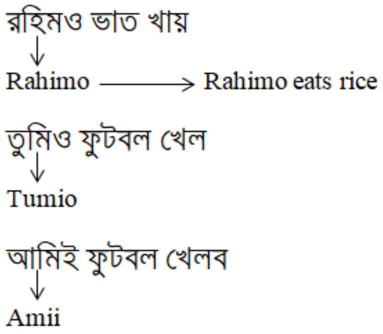


Figure 3.18: Faulty translation due to not identifying emphasizing tags

a full-name (Rahimo) due to the omission of checking for emphasizing tags. Similarly, the system misinterprets the subjects of other sentences in the figure as names (Tumio, Amii). This happens as the subject does not appear in our vocabulary due to having an emphasizing tag associated with it. Here, the translation of the first sentence actually should have been - "**Rahim also** eats rice", where the Bengali form of 'also' appears as an emphasizing tag (as suffix) not recognized in the translation presented in Figure 3.18

Hence, first, we need to check the suffix of the subject (name or pronoun) for any such tags in Bengali sentences. If we can identify any such tag, we need to separate it from the subject. Figure 3.19 illustrates the process of separating emphasizing tags with three different examples. After this separation, we can translate the name as discussed earlier, and take care of the emphasizing tags (suffixes) separately as shown in Figure 3.20.

Although we can apply this mechanism for name identification with emphasizing tags in our

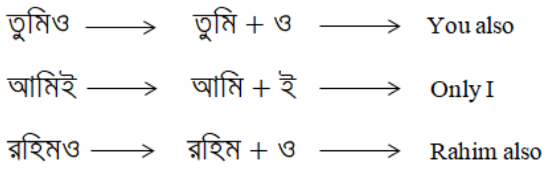


Figure 3.19: Separating emphasizing tags from subjects

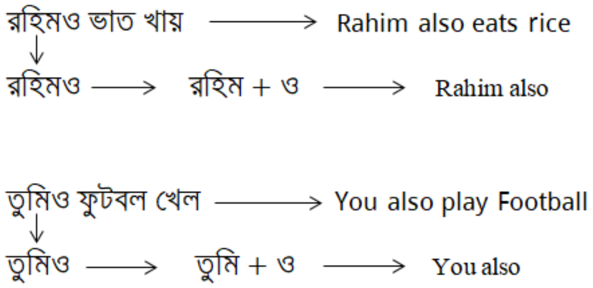


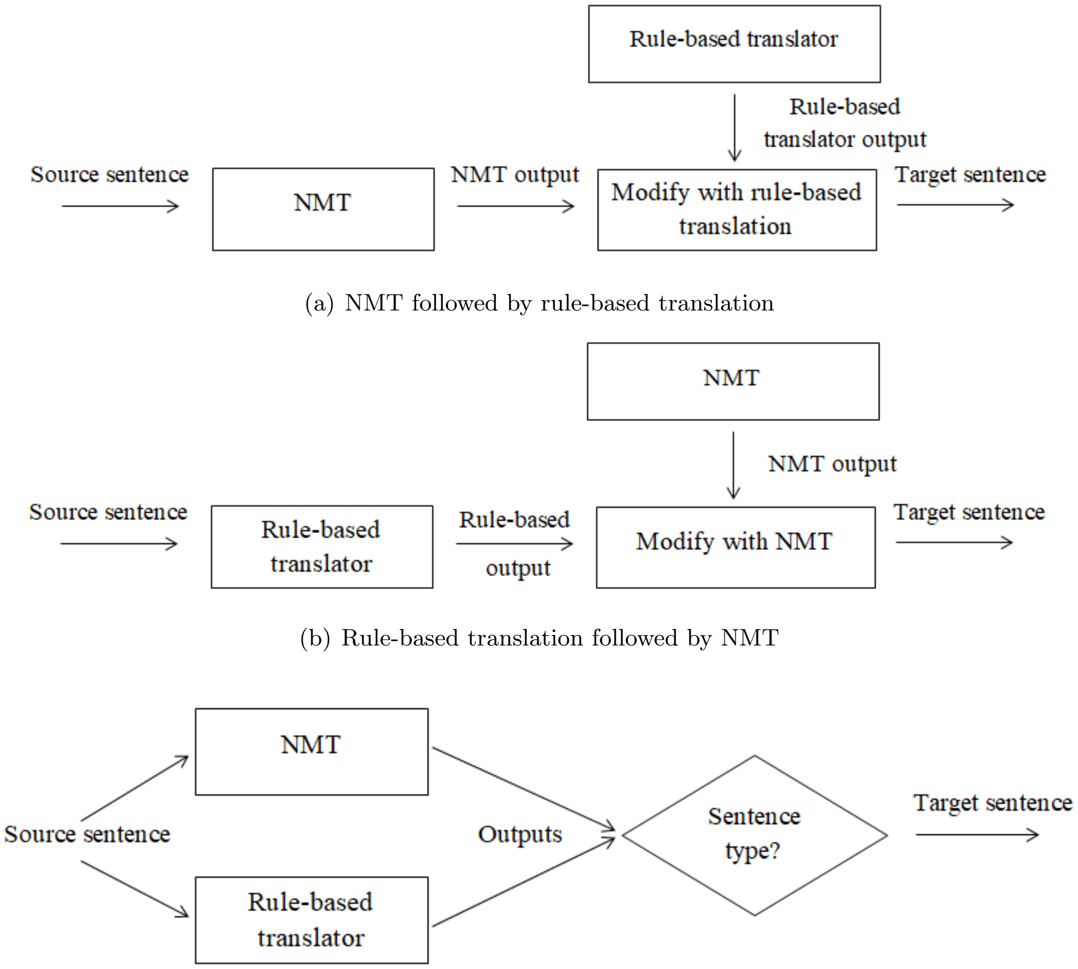
Figure 3.20: Separating emphasizing tags from subjects and corresponding translations

system appropriately, this may not generate the desired result in all cases. This happens as the forms of emphasizing tags can also be parts of actual names in some cases. We will focus on this point later with relevant examples in the next chapter. However, ignoring such possibilities of faulty identifications of emphasizing tags only in a limited number of cases, we can apply the proposed mechanism for name identification with emphasizing tags effectively in most of the cases.

3.4 Blending Rule-based Translator with NMT

Our proposed rule-based translator exhibits good performance in case of smaller sentences. This scope of rule-based translator expands as we add more rules continuously. However, it is near-to-impossible to implement unlimited and ever changing grammatical rules for any language. Besides, it is hard to deal with rule interactions in big systems [56], grammatical ambiguities [57], and idiomatic expressions [58]. Therefore, the potential of machine translation comes to light. Machine translation (MT) is a sub-field of computational linguistics that investigates the use of software to translate text or speech from one language to another. Recently, neural machine translation (NMT) has emerged as the most popular MT system since NMT is used in translation purpose by reputed organisations such as Google and Microsoft.

We have already discussed Google's Neural Machine Translation (GNMT) system in the previous



(a) NMT followed by rule-based translation

(b) Rule-based translation followed by NMT

(c) Either NMT or rule-based translation depending on the type of sentence

Figure 3.21: Different blending techniques between rule-based translation and NMT as explored in this study

chapter. GNMT works considerably well for translating between any pair of popular languages. However, NMT has its own major limitation in terms of generating accurate translations as shown in Figure 2.1 earlier. Thus, both rule-based translator and NMT exhibit their advantages and limitations compared to other. This finding leads to our next investigation on blending between rule-based translator and NMT.

To do so, we implement the classical NMT in our system from an open source resource [50]. Then, we integrate our proposed rule-based translator with the classical NMT to investigate whether such an integration can achieve a better performance in translation. We can explore the blending in three different ways:

- NMT followed by rule-based translation,
- Rule-based translation followed by NMT, and
- Either NMT or rule-based translation depending on the type of sentence.

Figure 3.21 illustrates how we can implement the possible blending approaches in our system. Besides, we present our blending approaches in Algorithm 1, 2, and 3. We discuss each of these three techniques in the next subsections.

3.4.1 NMT Followed by Rule-based Translation

Classical NMT initially requires training with parallel corpus (sentence pairs of source language and target language). In our case, we develop and adopt parallel corpus of different sizes containing Bengali-English sentence pairs for training the NMT. After training, we feed the intended input sentences to the NMT and generate the output sentences translated in English using the classical

Algorithm 1 Blending between rule-based translator and data-driven translator (NMT or SMT)

```
procedure GetBlendingOutput
Input : Source sentence (Bengali)
Output : Target sentence (English) after blending
    NMT_Outout ← Output generated by NMT (or SMT)
    RB_Output ← Output generated by rule-based translator
    Word ← An object with two attributes - token (word) and PoS_tag (Parts of Speech)
    PoS_tagger(sentence) ← ArrayList of Word objects with PoS_tag for each word in the sentence
    src_len ← Length of source sentence
    NMT_words ← ArrayList of Word objects (words with PoS tagging) in NMT
    RB_words ← ArrayList of Word objects (words with PoS tagging) in rule-based translation
    Translation_NMT+RB ← ArrayList of Word objects after blending (NMT followed by rule-based)
    Translation_RB+NMT ← ArrayList of Word objects after blending (RB followed by rule-based)
    Translation_NMTorRB ← ArrayList of Word objects after blending (Either NMT or rule-based)
    $NMT\_words := PoS\_tagger(NMT\_Output)$
    $RB\_words := PoS\_tagger(RB\_Output)$
    $Translation\_NMT + RB := PerformBlending(NMT\_words, RB\_words)$
    $Translation\_RB + NMT := PerformBlending(RB\_words, NMT\_words)$
    $Translation\_NMTorRB := PerformBlending\_NMTorRB(NMT\_Output, RB\_Output, src\_len)$
    display Translation_NMT+RB
    display Translation_RB+NMT
    display Translation_NMTorRB
```

NMT approach. In our experimentation, we consider a deep multi-layer recurrent neural network (RNN), which is unidirectional and uses LSTM as a recurrent unit [50].

After getting the NMT generated translated sentence, our blending approach applies grammatical rules on the translated sentence to further modify the sentence to improve its translation accuracy (Figure 3.21(a)). Algorithm 1 shows the skeleton of our blending approaches. Here, as discussed earlier, our system first tokenizes the source sentence to form the token-tagging table for rule-based translation. Using these token tagging information, our blending system can now substitute some of the words or phrases in the NMT generated translated sentences with the translated words obtained from our rule-based translator. More specifically, rule-based translator just further ameliorates the skeleton of the translated sentence that NMT has already built as shown in Algorithm 2.

Algorithm 2 Blending Module for NMT followed by rule-based approach and rule-based followed by NMT approach

```
procedure PerformBlending
Input: sentence1 ← ArrayList of Word objects in the sentence on which blending will be performed
Input: sentence2 ← ArrayList of Word objects in the sentence with which sentence1 will be blended
Output: Translated sentence after performing blending
    sent1_len ← Length of sentence1
    sent2_len ← Length of sentence2
    sent1_word ← A Word object in sentence1
    sent2_word ← A Word object in sentence2
    blended_Translation ← Output sentence after performing blending
    blended_Translation := NULL
    for i := o to i < sent1_len do
        sent1_word := sentence1.get(i)
        for j := 0 to j < sent2_len do
            sent2_word := sentence2.get(j)
            if sent1_word.token ≠ sent2_word.token then
                if sent1_word.PoS_tag = sent2_word.PoS_tag then
                    sentence1.set(i, sentence2.get(j))
                    sentence2.remove(j)
                    break
        blended_Translation := blended_Translation + " " + sentence1.get(i).token
    return blended_Translation
```

Algorithm 2 considers NMT generated translation and rule-based translation as 'sentence1' and 'sentence2' respectively for 'NMT followed by rule-based' blending approach. Here, if our blending system finds any pair of unmatched words (token) having the same parts-of-speech (PoS_tag), then our system replaces the NMT word with the corresponding rule-based word. This is how our system checks each word in the NMT generated translation with each word in the rule-based translation for replacement. Figure 3.22 shows an example of how this blending technique works. Here, apart from generating translation by NMT, we also generate its rule-based translation. However, NMT translation forms the *skeleton* of the translated sentence. Next, our blending system matches translations from both the translators token by token using the token tagging table of the rule-based translator. If the system finds any mismatch in any position or parts of speech then it replaces the NMT generated word in that position by the rule-based generated word exactly in the same position. Here, NMT translates Bengali name “oishee” to “Ishii” where “Ishii” takes the position of noun. However, “Oishee” takes the same position in rule-based translation. Therefore, first, this blending technique replaces “Ishii” by “Oishee” in the final translation. Afterwards, our system also replaces “had”, “finish”, and “his” by “was”, “finishing”, and “her” respectively keeping the words in other positions intact.

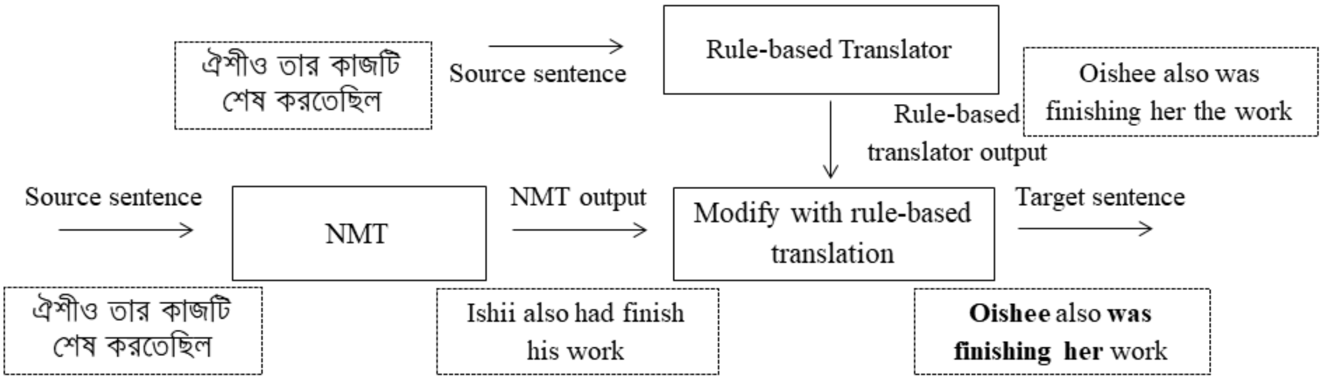

ঐশীও তার কাজটি শেষ করতেছিল
Source sentence
Rule-based Translator
Oishee also was finishing her the work
Rule-based translator output
Source sentence
NMT
NMT output
Modify with rule-based translation
Target sentence
ঐশীও তার কাজটি শেষ করতেছিল
Ishii also had finish his work
Oishee also was finishing her work

Figure 3.22: An example of NMT followed by rule-based translation

This technique proves itself to be the best blending technique so far, which we will illustrate in our experimental evaluation part later in this paper. The main reason behind this point is the fact that this technique realizes skeleton of translation from NMT and word-based attributes (such as person, number, tense, etc.) from rule-based translation. These two different forms of realizations best fit to strengths of the two different translation approaches.

3.4.2 Rule-based Translation Followed by NMT

This is another blending technique, which we propose and investigate. Here, we implement a reverse sequence of the previous blending technique. First, we pass the source sentence to the rule-based translator. Next, we further modify the translated sentence by NMT in this blending system as shown in Figure 3.21(b).

Similar to the earlier case, Algorithm 2 also illustrates this blending technique. This time, our system considers rule-based translation as 'sentence1' and NMT generated translation as 'sentence2' in Algorithm 2. Major limitation of this technique is that NMT runs completely on its own. NMT can generate completely wrong words in different positions in a sentence during translation since NMT always predicts the next word in sequence using probabilities. Its performance largely depends on the magnitude of training data. On the other hand, rule-based translator at least cannot pick wrong words since it only searches the vocabulary for any particular word translation and pick the translated word if found.

Therefore, if this blending system further modifies the rule-based translated sentence by NMT then it can happen that translation performance degrades in many cases. Only luck with this approach is when our rule-based translator cannot recognize the source sentence due to lack of appropriate rule-set. This point relates to one of the most common advantages of using machine translation - NMT works better for translating any random sentence (not all sentences needs to be covered by rules), and for fast and cheap translation. Figure 3.23 presents an example on how this technique performs translation.

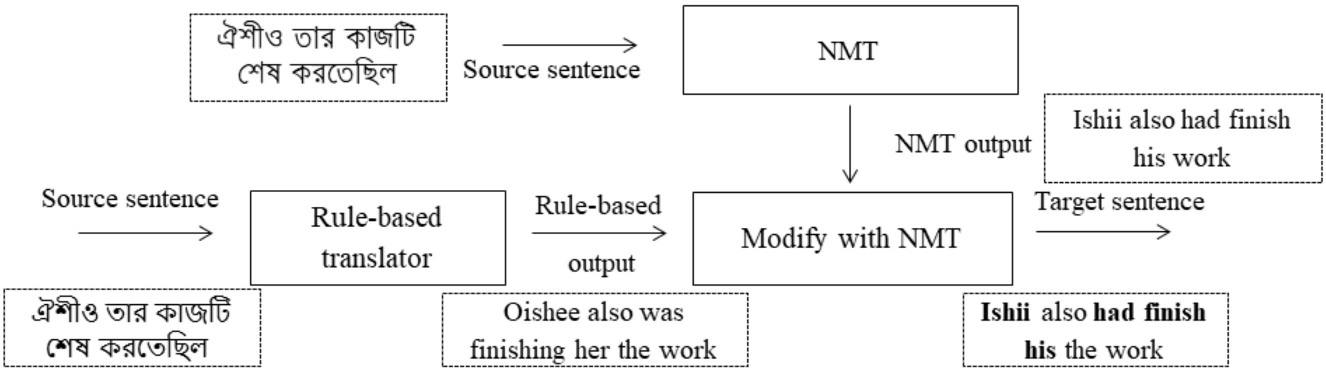

ঐশীও তার কাজটি শেষ করতেছিল
Source sentence
NMT
NMT output
Ishii also had finish his work
Source sentence
Rule-based translator
Rule-based output
Modify with NMT
Target sentence
ঐশীও তার কাজটি শেষ করতেছিল
Oishee also was finishing her the work
Ishii also had finish his the work

Figure 3.23: An example of rule-based translation followed by NMT

Here, initially, the two unmatched words, "Oishee" in rule-based and "Ishii" in NMT, hold the same position in the translated sentences. Therefore, first, this blending approach replaces "Oishee" by "Ishii". Afterwards, our system also replaces "was", "finishing", and "her" by "had", "finish", and "his" respectively as shown in Figure 3.23.

3.4.3 Either NMT or Rule-based Translator

This blending technique is much simpler compared to earlier ones. It performs choosing one between two translations generated by rule-based translator and NMT separately as shown in Figure 3.21(c). However, this blending system needs to make the choice based on some criteria so that it chooses the better one.

In our system, rule-based translator works better for small sentences. More specifically, our rule-based translation system implements rules for the sentences having smaller length (not more than 7 words) and simpler structure so far. As we keep adding more rules, the scope of translation will definitely grow for rule-based translator. Therefore, this blending approach chooses rule-based translation if the source sentence is smaller in length. Otherwise, it chooses NMT generated translation as the output translation. We present this blending approach in Algorithm 3.

Algorithm 3 Blending module for Either NMT or rule-based approach

```
procedure PerformBlending_NMTorRB
Input : sentence1 ← Translation generated by NMT
Input : sentence2 ← Translation generated by rule-based translator
Input : source_length ← Length of source sentence
Output : Translated sentence after performing blending
    blended_Translation ← Output sentence after performing blending
    blended_Translation := NULL
    if source_length $\leq$ 7 then
        blended_Translation := sentence2 // rule-based translation
    else
        blended_Translation := sentence1 // NMT
    return blended_Translation
```

Figure 3.24 shows a working example of this blending technique. In the figure, we identify the source sentence as a small sentence with only five words. Our blending system considers sentences consisting of less than 8 words as small sentences. Therefore, the system selects translation generated by the rule-based translator as the final translation and ignores NMT this time. Besides, note that we can update the selection criteria (sentence type) in this blending system according to the scope of the rule-based translator. The more we add rules, the more types of sentences we can translate using rule-based translator. Therefore, selection criteria can be made much more flexible and tricky in this system depending on performance analysis after incorporating more rules.

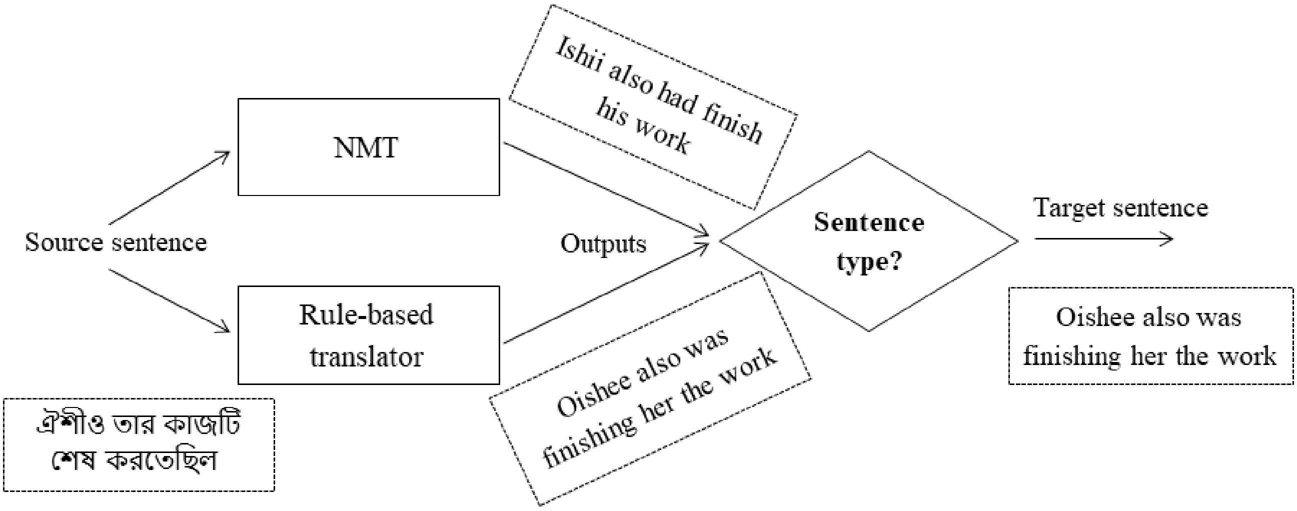

NMT
Ishii also had finish his work
Source sentence
Outputs
Sentence type?
Target sentence
Rule-based translator
ঐশীও তার কাজটি শেষ করতেছিল
Oishee also was finishing her the work
Oishee also was finishing her the work

Figure 3.24: An example of choosing either NMT or rule-based translation

Chapter 4

Performance Evaluation

We perform rigorous performance evaluation of our different approaches on the basis of different types of metrics. In this chapter, we present our experimental settings, data sets, performance metrics, and all kinds of results along with corresponding analyses.

4.1 Experimental Settings

We need to employ considerable resources for our experimentation, as such experimentation are resource-hungry and time consuming in general. We present the resources and settings we utilized in our experiments in the next subsection.

4.1.1 Settings for Experimentation of Rule-based Translator

We use software and hardware resources for implementing our rule-based translation system. Here, we use JAVA language, Netbeans platform/IDE, Sqlite database, Opennlp tools [49], and Windows 10 (64 - bit) operating system as software resources. Besides, we use Core i3 - 2310M (2.10 GHz) processor, 4.00 GB RAM, and 1 TB HDD hard disk as hardware resources.

We face a major challenge regarding taking input and parsing Bengali texts in Java. To do so, initially, we set the text encoding in Netbeans to UTF-8 [15]. However, Bengali fonts and texts do not appear in Netbeans properly. Afterwards, we change the font settings (font family, font size, etc.) to finally be able to work with Bengali texts in Netbeans successfully. Besides, integration of Sqlite library [15] with Netbeans IDE for connecting database is another important feature of our rule-based translator. Since we need a Bengali to English dictionary in the system, we require a database (vocabulary) to retrieve the Bengali to English word translation. Additionally, for keeping other information such as token-tagging table, number table, person table, etc., we need to connect a database with our system. For this purpose, we integrate Sqlite with Netbeans by adding a jar file for Sqlite.

Regarding Bengali to English dictionary, we do not find any well-defined dictionary format so that we can import it to our system's database directly. Therefore, we ourselves insert a reasonable amount of words in our database.

4.1.2 Setting of Experimentation with NMT

To perform our experimentation with NMT, we utilize TensorFlow in our system. Specifically, we install TensorFlow version 1.4.2 using Python's pip package manager having Ubuntu 16.04 as the operating system. We pull the source code of NMT from Github to our system by running the command - "git clone https://github.com/tensorflow/nmt/". Here, we use Python language, PyCharm platform/IDE, Tensorflow library, and Linux (64 - bit) operating system. Besides, we use Core i3 - 2310M (2.10 GHz) processor, 4.00 GB RAM, and 1 TB HDD hard disk as hardware resources.

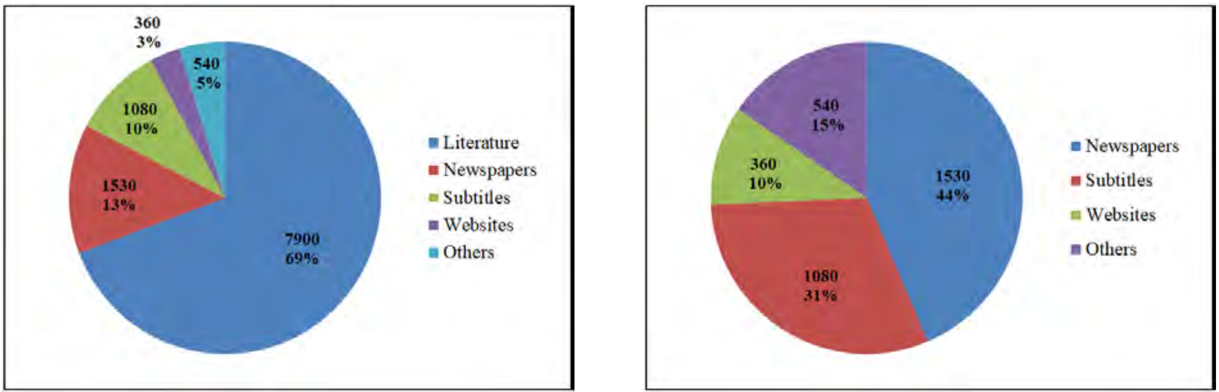
To start the experimentation, we design datasets for training NMT and testing its performance. We use the following hyper-parameters in our system for training NMT with our designed datasets - 1) 12000 training steps, 2) 2 hidden layers, 3) 20% dropout rate, and 4) 100 steps per statistics. We choose these hyper-parameters based on the benchmarks achieved for English-Vietnamese and German-English translation as claimed in [50].

4.2 Datasets

Designing and developing datasets has been one of the most challenging and time intensive tasks in our experimentation. For training the NMT reasonably, we require a large parallel corpus containing both source language and target language. In our case, NMT requires such a corpus of Bengali-English sentence pairs. However, we find very few sources available for constructing a reasonable sized dataset containing Bengali-English sentence pairs.

4.2.1 Demography of Datasets

We create the corpus at our own by translating different Bengali sentences to English one by one. We develop our dataset of Bengali-English parallel corpus from well-established contents such as Al-Quran [52], newspapers [53], movie subtitles [54], and university websites [55]. Besides, we translate different example-based individual Bengali sentences into English and accumulate them in the dataset. Figure 4.1(a) illustrates a demography of our full dataset. Initially, we experiment with only literature-based source (Al-Quran) of our full dataset since its size is large enough to be considered as a separate dataset when compared to the size our full dataset. Afterwards, we also experiment with our full dataset with an intent to generate results from a fairly diversified dataset. Therefore, our full dataset also includes another dataset (custom dataset) as its subset (excluding the literature-based dataset). However, we do not consider using this custom dataset independently in our experimentation since its size is too small to train an NMT system reasonably. We present a demography of our custom dataset (a subset of full dataset) in Figure 4.1(b). Additionally, we also perform translations over individual sentences and analyze their outcomes.



(a) Full dataset

(b) Custom dataset (subset of the full dataset)

Figure 4.1: Demography of our datasets

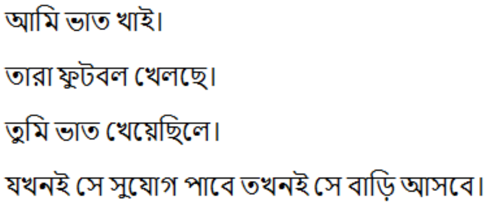
There is another dataset containing more than 1 million Bengali-English parallel sentences, which is made available in a website called ‘GlobalVoices’ [61]. However, the sentences in this dataset contain numerous unknown characters and words (even from other languages such as Arabic, Chinese, German, etc.), which needs to be cleaned first before using in experimentation. Therefore, we carefully remove such unknown characters from this dataset. Besides, there are English sentences that are not proper translations of corresponding Bengali sentences in this dataset. Therefore, this dataset requires rigorous manual checking and corrections for each sentence pair. Table 4.1 shows summary of the different datasets.

Dataset	Number of sentences	Sources	Used in experimentation?
Literature-based	8,000	Al-Quran	Yes
Custom	3,500	Newspaper, subtitles, websites, etc.	Blended in full dataset
Full (combined)	11,500	Literature-based and custom dataset	Yes
GlobalVoices	10,31,725	Website	Yes

Table 4.1: Summary of the different datasets

4.2.2 Individual Sentences

We design individual sentences mainly for testing the performance of our rule-based translator after integrating different rules. This requires having different categories of sentences from the source language. In our case, we consider 540 individual Bengali sentences for translation, which cover different categories (rules) of sentences. To do so, we collect 540 individual sentences. Figure 4.2 shows some examples on how we choose different categories of sentences. For example, the first sentence in the figure is an example of a simple present tense. The second sentence and the third sentence refer to present continuous tense and simple past tense respectively. The last sentence is an example of a complex sentence.



আমি ভাত খাই।

তারা ফুটবল খেলছে।

তুমি ভাত খেয়েছিলে।

যখনই সে সুযোগ পাবে তখনই সে বাড়ি আসবে।

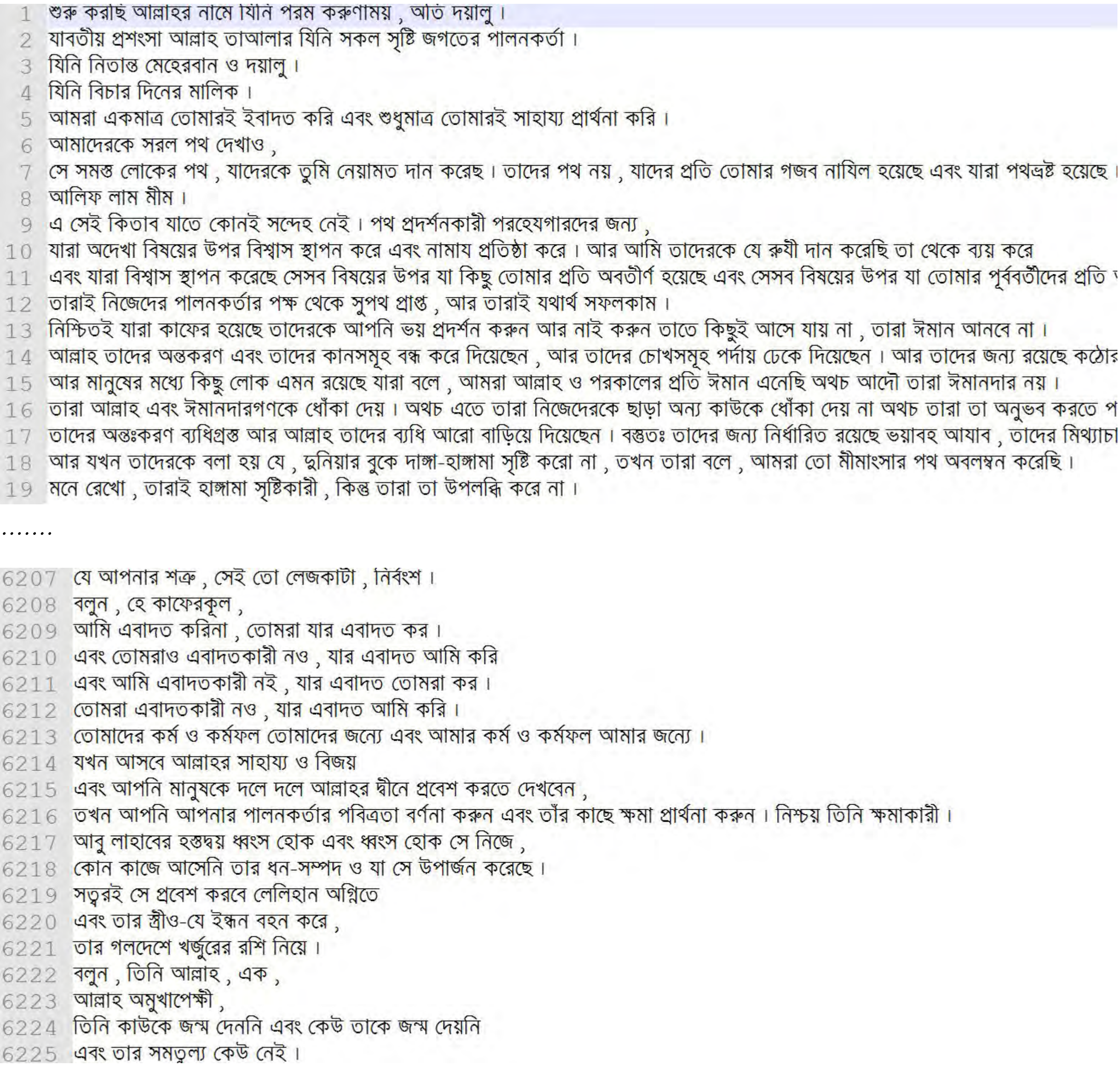
Figure 4.2: Individual sentences for evaluating translations by our rule-based translator

4.2.3 Literature-based Dataset

Unlike rule-based translator, we require a large parallel corpus of Bengali-English sentence pairs in NMT. Therefore, we develop our literature-based dataset keeping NMT as the prime focus. It is a challenging task to collect and compile a large parallel corpus using Bengali literature, as most of the translations of Bengali literature books are available as scanned copies that are not editable. In this regard, we find Al-Quran (the holy Islamic book) to be available as parallel corpus consisting of Bengali-English sentence pairs. Therefore, we adopt Al-Quran as a source of our literature-based dataset, which contains around 8,000 Bengali-English sentence pairs. Figure 4.3 and Figure 4.4 show snippets of our Bengali and English datasets respectively extracted in this manner. Note that it is not ideal to consider only Al-Quran as a source for Bengali-English translation for two reasons - 1) Most of the source sentences are tough for a machine (or even for a human) to realize and process, and 2) Translations are relatively complex here to some extent.

In addition to our dataset, we need to provide vocabulary files of both source and target languages for predicting words during generating translations by NMT. There are two separate vocabulary files for Bengali and English, which we generate from Bengali and English sentences respectively. These files contain one unique token (word) per line. Besides, NMT needs the words to be sorted in descending order according to their frequency (number of appearances) in the whole corpus. Figure 4.5 and Figure 4.6 show snippets of vocabulary files of Bengali and English respectively.

Another point is that each vocabulary file should begin with three special tokens as shown in Figure 4.7. Here, 1) “unk” refers to replacing the unknown word translations as “unk”, 2) “s” refers to a starting symbol “s” (“tgt_sos_id” in our code) enabling the decoding (translation) process to be started as soon as the decoder receives this symbol, and 3) “/s” refers to an output symbol (“tgt_eos_id” in our code) enabling the translation process to be continued until this end-of-sentence marker “/s”.

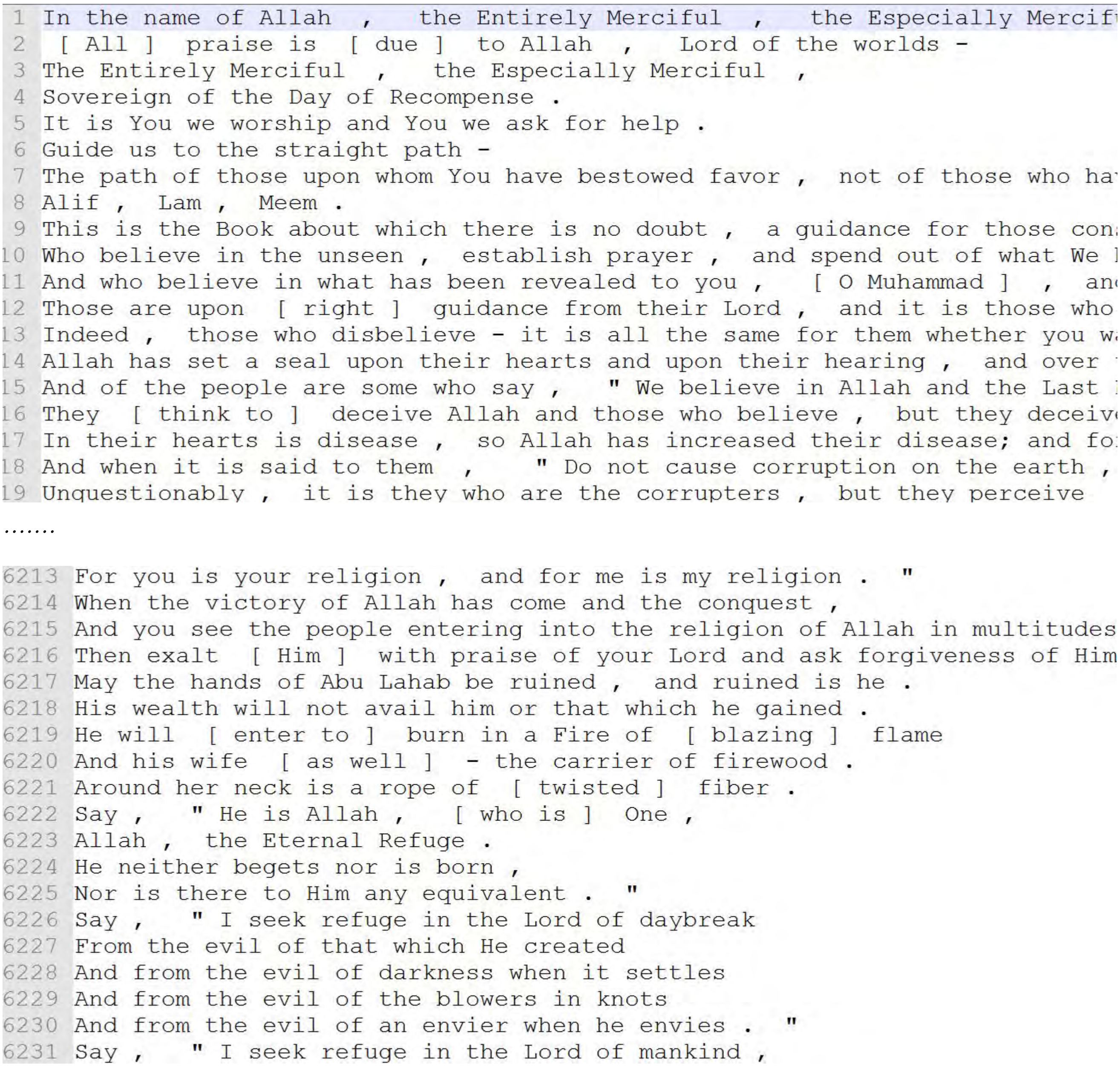


```
1  শুরু করছি আল্লাহর নামে যিনি পরম করুণাময় , অতি দয়ালু ।
2  যাবতীয় প্রশংসা আল্লাহ তাআলার যিনি সকল সৃষ্টি জগতের পালনকর্তা ।
3  যিনি নিতান্ত মেহেরবান ও দয়ালু ।
4  যিনি বিচার দিনের মালিক ।
5  আমরা একমাত্র তোমারই ইবাদত করি এবং শুধুমাত্র তোমারই সাহায্য প্রার্থনা করি ।
6  আমাদেরকে সরল পথ দেখাও ,
7  সে সমস্ত লোকের পথ , যাদেরকে তুমি নেয়ামত দান করেছ । তাদের পথ নয় , যাদের প্রতি তোমার গজব নাযিল হয়েছে এবং যারা পথভ্রষ্ট হয়েছে ।
8  আলিফ লাম মীম ।
9  এ সেই কিতাব যাতে কোনই সন্দেহ নেই । পথ প্রদর্শনকারী পরহেযগারদের জন্য ,
10 যারা অদেখা বিষয়ের উপর বিশ্বাস স্থাপন করে এবং নামায প্রতিষ্ঠা করে । আর আমি তাদেরকে যে রুযী দান করেছি তা থেকে ব্যয় করে
11 এবং যারা বিশ্বাস স্থাপন করেছে সেসব বিষয়ের উপর যা কিছু তোমার প্রতি অবতীর্ণ হয়েছে এবং সেসব বিষয়ের উপর যা তোমার পূর্ববর্তীদের প্রতি
12 তারাই নিজেদের পালনকর্তার পক্ষ থেকে সুপথ প্রাপ্ত , আর তারাই যথার্থ সফলকাম ।
13 নিশ্চিতই যারা কাফের হয়েছে তাদেরকে আপনি ভয় প্রদর্শন করুন আর নাই করুন তাতে কিছুই আসে যায় না , তারা ঈমান আনবে না ।
14 আল্লাহ তাদের অন্তকরণ এবং তাদের কানসমূহ বন্ধ করে দিয়েছেন , আর তাদের চোখসমূহ পর্দায় ঢেকে দিয়েছেন । আর তাদের জন্য রয়েছে কঠোর
15 আর মানুষের মধ্যে কিছু লোক এমন রয়েছে যারা বলে , আমরা আল্লাহ ও পরকালের প্রতি ঈমান এনেছি অথচ আদৌ তারা ঈমানদার নয় ।
16 তারা আল্লাহ এবং ঈমানদারগণকে ধোঁকা দেয় । অথচ এতে তারা নিজেদেরকে ছাড়া অন্য কাউকে ধোঁকা দেয় না অথচ তারা তা অনুভব করতে প
17 তাদের অন্তঃকরণ ব্যধিগ্রস্ত আর আল্লাহ তাদের ব্যধি আরো বাড়িয়ে দিয়েছেন । বস্তুতঃ তাদের জন্য নির্ধারিত রয়েছে ভয়াবহ আযাব , তাদের মিথ্যাচা
18 আর যখন তাদেরকে বলা হয় যে , দুনিয়ার বুকে দাঙ্গা-হাঙ্গামা সৃষ্টি করো না , তখন তারা বলে , আমরা তো মীমাংসার পথ অবলম্বন করেছি ।
19 মনে রেখো , তারাই হাঙ্গামা সৃষ্টিকারী , কিন্তু তারা তা উপলব্ধি করে না ।
```

....

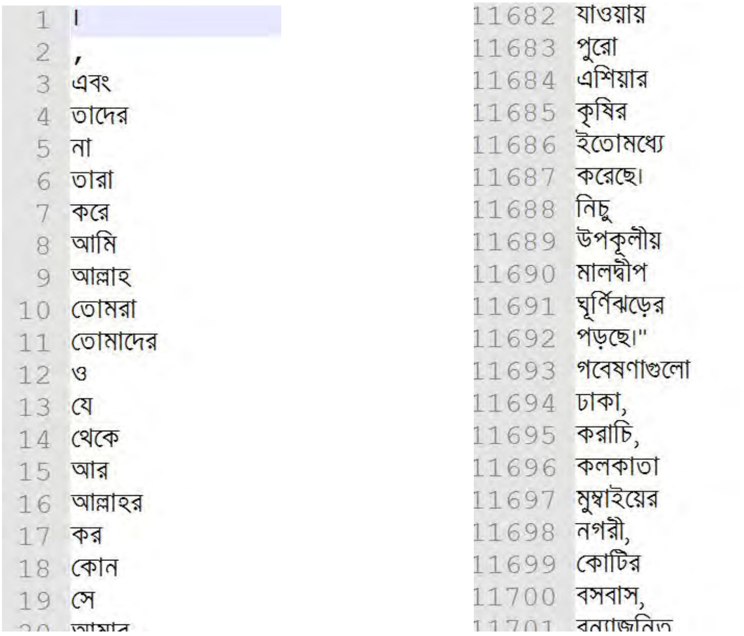
```
6207 যে আপনার শত্রু , সেই তো লেজকাটা , নির্বংশ ।
6208 বলুন , হে কাফেরকূল ,
6209 আমি এবাদত করিনা , তোমরা যার এবাদত কর ।
6210 এবং তোমরাও এবাদতকারী নও , যার এবাদত আমি করি
6211 এবং আমি এবাদতকারী নই , যার এবাদত তোমরা কর ।
6212 তোমরা এবাদতকারী নও , যার এবাদত আমি করি ।
6213 তোমাদের কর্ম ও কর্মফল তোমাদের জন্যে এবং আমার কর্ম ও কর্মফল আমার জন্যে ।
6214 যখন আসবে আল্লাহর সাহায্য ও বিজয়
6215 এবং আপনি মানুষকে দলে দলে আল্লাহর দ্বীনে প্রবেশ করতে দেখবেন ,
6216 তখন আপনি আপনার পালনকর্তার পবিত্রতা বর্ণনা করুন এবং তাঁর কাছে ক্ষমা প্রার্থনা করুন । নিশ্চয় তিনি ক্ষমাকারী ।
6217 আবু লাহাবের হস্তদ্বয় ধ্বংস হোক এবং ধ্বংস হোক সে নিজে ,
6218 কোন কাজে আসেনি তার ধন-সম্পদ ও যা সে উপার্জন করেছে ।
6219 সত্বরই সে প্রবেশ করবে লেলিহান অগ্নিতে
6220 এবং তার স্ত্রীও-যে ইন্ধন বহন করে ,
6221 তার গলদেশে খর্জুরের রশি নিয়ে ।
6222 বলুন , তিনি আল্লাহ , এক ,
6223 আল্লাহ অমুখাপেক্ষী ,
6224 তিনি কাউকে জন্ম দেননি এবং কেউ তাকে জন্ম দেয়নি
6225 এবং তার সমতুল্য কেউ নেই ।
```

Figure 4.3: Partial Bengali literature-based dataset (extracted from Al-Quran)



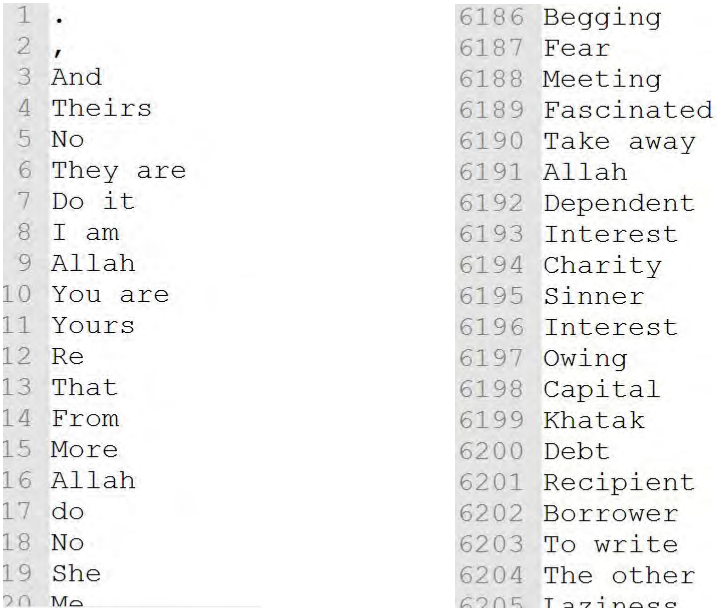
```
1  In the name of Allah ,  the Entirely Merciful ,  the Especially Mercif
2   [ All ]  praise is  [ due ]  to Allah ,  Lord of the worlds -
3  The Entirely Merciful ,  the Especially Merciful ,
4  Sovereign of the Day of Recompense .
5  It is You we worship and You we ask for help .
6  Guide us to the straight path -
7  The path of those upon whom You have bestowed favor ,  not of those who ha
8  Alif ,  Lam ,  Meem .
9  This is the Book about which there is no doubt ,  a guidance for those con
10 Who believe in the unseen ,  establish prayer ,  and spend out of what We
11 And who believe in what has been revealed to you ,  [ O Muhammad ]  ,  an
12 Those are upon  [ right ]  guidance from their Lord ,  and it is those who
13 Indeed ,  those who disbelieve - it is all the same for them whether you w
14 Allah has set a seal upon their hearts and upon their hearing ,  and over
15 And of the people are some who say ,  " We believe in Allah and the Last
16 They  [ think to ]  deceive Allah and those who believe ,  but they deceiv
17 In their hearts is disease ,  so Allah has increased their disease; and fo
18 And when it is said to them ,   " Do not cause corruption on the earth ,
19 Unquestionably ,  it is they who are the corrupters ,  but they perceive
.......

6213 For you is your religion ,  and for me is my religion .  "
6214 When the victory of Allah has come and the conquest ,
6215 And you see the people entering into the religion of Allah in multitudes
6216 Then exalt  [ Him ]  with praise of your Lord and ask forgiveness of Him
6217 May the hands of Abu Lahab be ruined ,  and ruined is he .
6218 His wealth will not avail him or that which he gained .
6219 He will  [ enter to ]  burn in a Fire of  [ blazing ]  flame
6220 And his wife  [ as well ]  - the carrier of firewood .
6221 Around her neck is a rope of  [ twisted ]  fiber .
6222 Say ,  " He is Allah ,  [ who is ]  One ,
6223 Allah ,  the Eternal Refuge .
6224 He neither begets nor is born ,
6225 Nor is there to Him any equivalent .  "
6226 Say ,  " I seek refuge in the Lord of daybreak
6227 From the evil of that which He created
6228 And from the evil of darkness when it settles
6229 And from the evil of the blowers in knots
6230 And from the evil of an envier when he envies .  "
6231 Say ,  " I seek refuge in the Lord of mankind ,
```

Figure 4.4: Partial English literature-based dataset (extracted from Al-Quran)



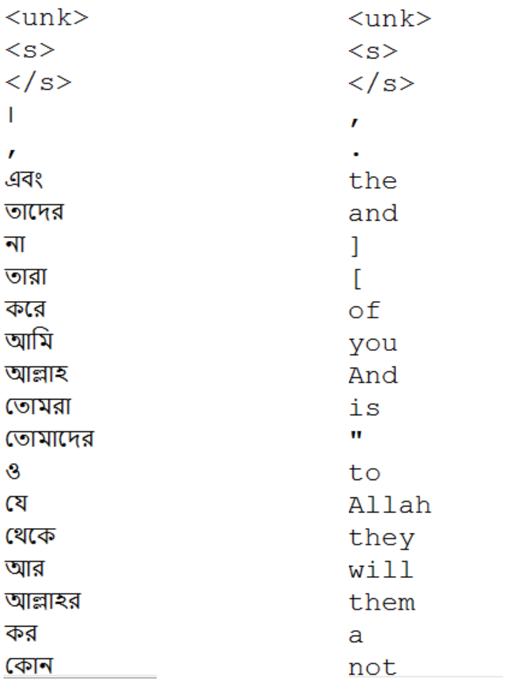
```
1  ।                 11682 যাওয়ায়
2  ,                 11683 পুরো
3  এবং               11684 এশিয়ার
4  তাদের             11685 কৃষির
5  না                11686 ইতোমধ্যে
6  তারা              11687 করেছে।
7  করে               11688 নিচু
8  আমি               11689 উপকূলীয়
9  আল্লাহ            11690 মালদ্বীপ
10 তোমরা             11691 ঘূর্ণিঝড়ের
11 তোমাদের           11692 পড়ছে।"
12 ও                 11693 গবেষণাগুলো
13 যে                11694 ঢাকা,
14 থেকে              11695 করাচি,
15 আর                11696 কলকাতা
16 আল্লাহর           11697 মুম্বাইয়ের
17 কর                11698 নগরী,
18 কোন               11699 কোটির
19 সে                11700 বসবাস,
```

Figure 4.5: Partial Bengali vocabulary



```
1  .                 6186 Begging
2  ,                 6187 Fear
3  And               6188 Meeting
4  Theirs            6189 Fascinated
5  No                6190 Take away
6  They are          6191 Allah
7  Do it             6192 Dependent
8  I am              6193 Interest
9  Allah             6194 Charity
10 You are           6195 Sinner
11 Yours             6196 Interest
12 Re                6197 Owing
13 That              6198 Capital
14 From              6199 Khatak
15 More              6200 Debt
16 Allah             6201 Recipient
17 do                6202 Borrower
18 No                6203 To write
19 She               6204 The other
```

Figure 4.6: Partial English vocabulary



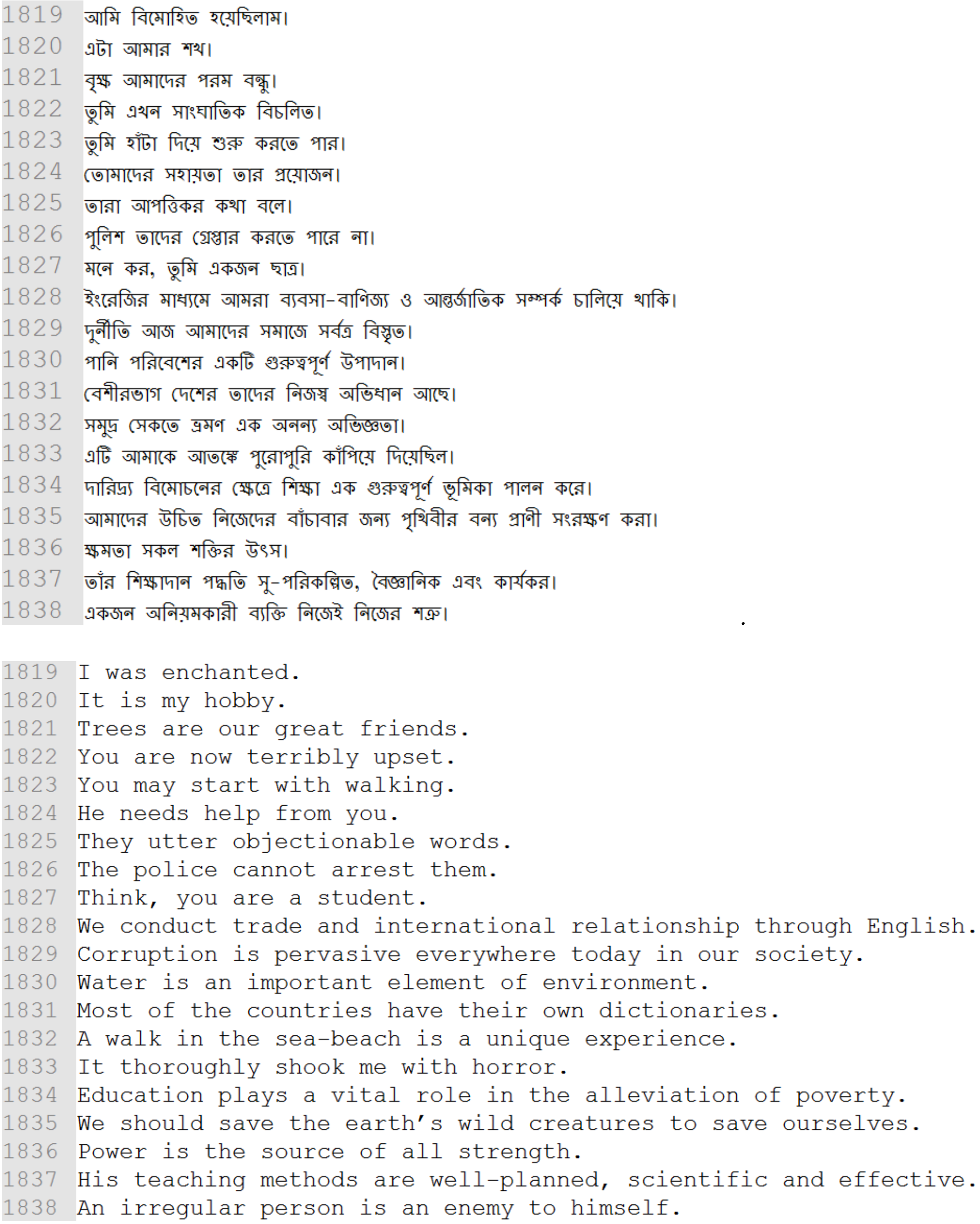
<unk>	<unk>
<s>	<s>
</s>	</s>
।	,
,	.
এবং	the
তাদের	and
না	]
তারা	[
করে	of
আমি	you
আল্লাহ	And
তোমরা	is
তোমাদের	"
ও	to
যে	Allah
থেকে	they
আর	will
আল্লাহর	them
কর	a
কোন	not

Figure 4.7: Beginning of vocabulary files

4.2.4 Custom Dataset

Literature-based dataset from Al-Quran has a considerable size (around 8,000 sentence pairs) that can be used for training the NMT. However, sentences of Al-Quran might not be considered as standard enough to represent a language. Nonetheless, Al-Quran can also be quite complex for a machine to recognize and process. Therefore, we develop another dataset with more usual and realizable sentences representing both the source and the target languages for the purpose of training the NMT better.

Major sources of this custom dataset are newspaper articles, movie subtitles, websites, etc. However, we cannot import any of such existing sources directly as Bengali-English parallel corpus. Each source (Bengali) sentence from the newspapers or the subtitles requires manual checking and editing in generating the parallel target (English) sentence. We perform this at our own to develop the custom dataset, which is presented in Appendix. Size of our custom dataset is around 3,500 parallel sentences, which is so small that it cannot train an NMT system reasonably. Therefore, we do not



```
1819 আমি বিমোহিত হয়েছিলাম।
1820 এটা আমার শখ।
1821 বৃক্ষ আমাদের পরম বন্ধু।
1822 তুমি এখন সাংঘাতিক বিচলিত।
1823 তুমি হাঁটা দিয়ে শুরু করতে পার।
1824 তোমাদের সহায়তা তার প্রয়োজন।
1825 তারা আপত্তিকর কথা বলে।
1826 পুলিশ তাদের গ্রেপ্তার করতে পারে না।
1827 মনে কর, তুমি একজন ছাত্র।
1828 ইংরেজির মাধ্যমে আমরা ব্যবসা-বাণিজ্য ও আন্তর্জাতিক সম্পর্ক চালিয়ে থাকি।
1829 দুর্নীতি আজ আমাদের সমাজে সর্বত্র বিস্তৃত।
1830 পানি পরিবেশের একটি গুরুত্বপূর্ণ উপাদান।
1831 বেশীরভাগ দেশের তাদের নিজস্ব অভিধান আছে।
1832 সমুদ্র সেকতে ভ্রমণ এক অনন্য অভিজ্ঞতা।
1833 এটি আমাকে আতঙ্কে পুরোপুরি কাঁপিয়ে দিয়েছিল।
1834 দারিদ্র্য বিমোচনের ক্ষেত্রে শিক্ষা এক গুরুত্বপূর্ণ ভূমিকা পালন করে।
1835 আমাদের উচিত নিজেদের বাঁচাবার জন্য পৃথিবীর বন্য প্রাণী সংরক্ষণ করা।
1836 ক্ষমতা সকল শক্তির উৎস।
1837 তাঁর শিক্ষাদান পদ্ধতি সু-পরিকল্পিত, বৈজ্ঞানিক এবং কার্যকর।
1838 একজন অনিয়মকারী ব্যক্তি নিজেই নিজের শত্রু।
```

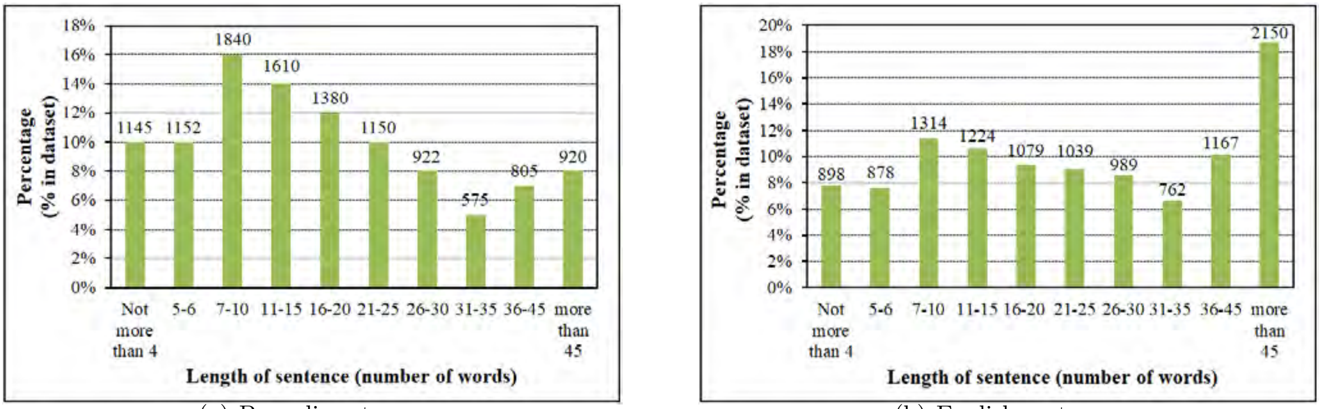
```
1819 I was enchanted.
1820 It is my hobby.
1821 Trees are our great friends.
1822 You are now terribly upset.
1823 You may start with walking.
1824 He needs help from you.
1825 They utter objectionable words.
1826 The police cannot arrest them.
1827 Think, you are a student.
1828 We conduct trade and international relationship through English.
1829 Corruption is pervasive everywhere today in our society.
1830 Water is an important element of environment.
1831 Most of the countries have their own dictionaries.
1832 A walk in the sea-beach is a unique experience.
1833 It thoroughly shook me with horror.
1834 Education plays a vital role in the alleviation of poverty.
1835 We should save the earth’s wild creatures to save ourselves.
1836 Power is the source of all strength.
1837 His teaching methods are well-planned, scientific and effective.
1838 An irregular person is an enemy to himself.
```

Figure 4.8: Partial Custom dataset

use this dataset independently in our experimentation. We present a snippet of our custom dataset in Figure 4.8.

4.2.5 Full Dataset

Our literature-based dataset consists of sentences only from the holy Al-Quran, whereas our custom dataset is not large enough to be considered for training an NMT system. Therefore, we combine our custom dataset with our literature-based dataset for experimenting with a larger and more diversified dataset. Thus, our full dataset (combined with literature-based dataset and custom dataset) consists of around 11,500 Bengali-English sentence pairs from different sources such as Al-Quran, newspaper articles, movie subtitles, university websites, etc. Besides, both Bengali and English sentences in our full dataset vary in size or length. Figure 4.9 reflects percentages (%) of different types of sentences in our full dataset in terms of different sizes or lengths. In addition to that, we also generate necessary vocabulary files for our full dataset similar to what we have done in the case of literature-based dataset.



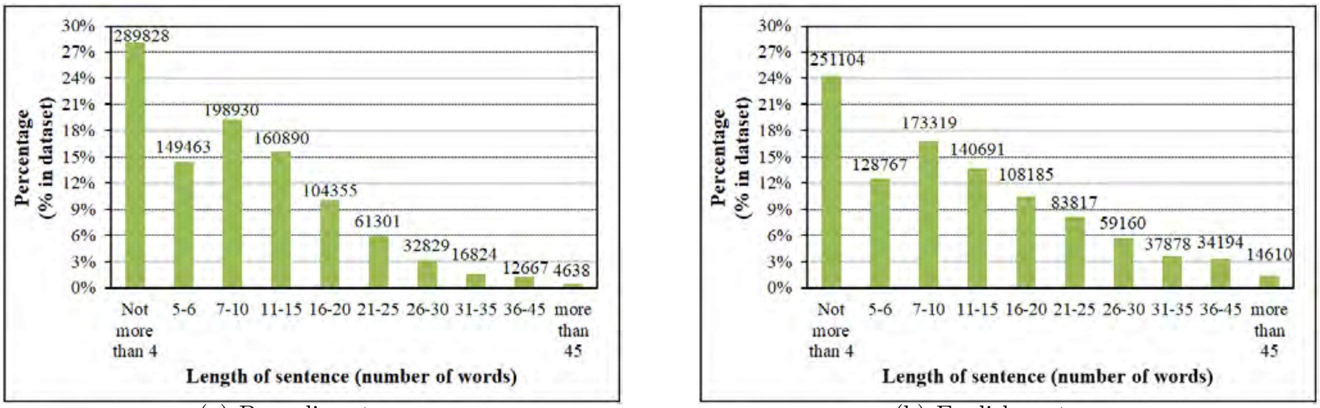
(a) Bengali sentences

(b) English sentences

Figure 4.9: Percentages of sizes of sentences in the full dataset

4.2.6 GlobalVoices Dataset

Data-driven translators (NMT or SMT) require significant amount of training data. However, our full dataset contains up to 11,500 parallel Bengali-English sentences, which represents the context of low-resource language. Therefore, we develop a larger Bengali-English parallel corpus containing more than one million sentence pairs to extend our experimentation to a high-resource context. Figure 4.10 reflects percentages (%) of different types of sentences in this dataset in terms of different sizes or lengths.



(a) Bengali sentences

(b) English sentences

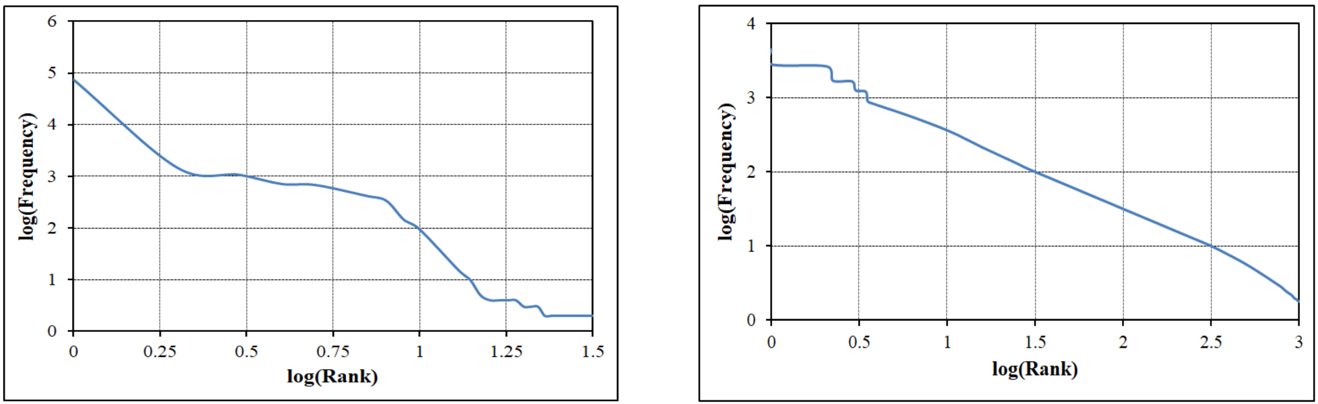
Figure 4.10: Percentages of sizes of sentences in the GlobalVoices dataset

4.2.7 Representativeness in Our Datasets

We analyze the representativeness of our dataset using Zipf's law [51]. Zipf's law pertains to frequency distribution of words in a language (or a dataset of the language, which is large enough to be a representative of the language). To illustrate Zipf's law, let we have a dataset and let there be V unique words in the dataset. For each word in the dataset, we compute how many times the word occurs in the dataset. We refer this as Freq(word). Then, we rank the words (Rank(word)) in descending order of their frequencies. Let r be the rank of a word and Prob(r) be the probability of a word at rank r. By definition, $Prob(r) = freq(r)/N$, where freq(r) is the number of times the word at rank r appears in the dataset. Besides, N is the total number of words in the dataset. Zipf's law states that $r \times Prob(r) = A$, where A is a constant that we should empirically determine from the dataset. Taking into account that $Prob(r) = freq(r)/N$, we can rewrite Zipf's law as $r \times freq(r) = A \times N$.

To demonstrate that Zipf's law holds in our dataset, we compute freq(r) that involves computing frequency and ranking of each word. Then, we compute $r \times freq(r)$ to check whether $r \times freq(r)$ becomes approximately a constant in all cases. The simplest way to show that Zipf's law holds in a dataset is to plot the computed values and check whether the slope is proportionately downward. Here, instead of plotting freq(r) versus rank, it is better to plot log(r) in the X axis and log(freq(r)) in the Y axis [51]. Accordingly, we plot the computed values for both Bengali corpus and English corpus separately in two different graphs.

We present the graphs for our first dataset (literature-based dataset) in Figure 4.11(a) and Fig-

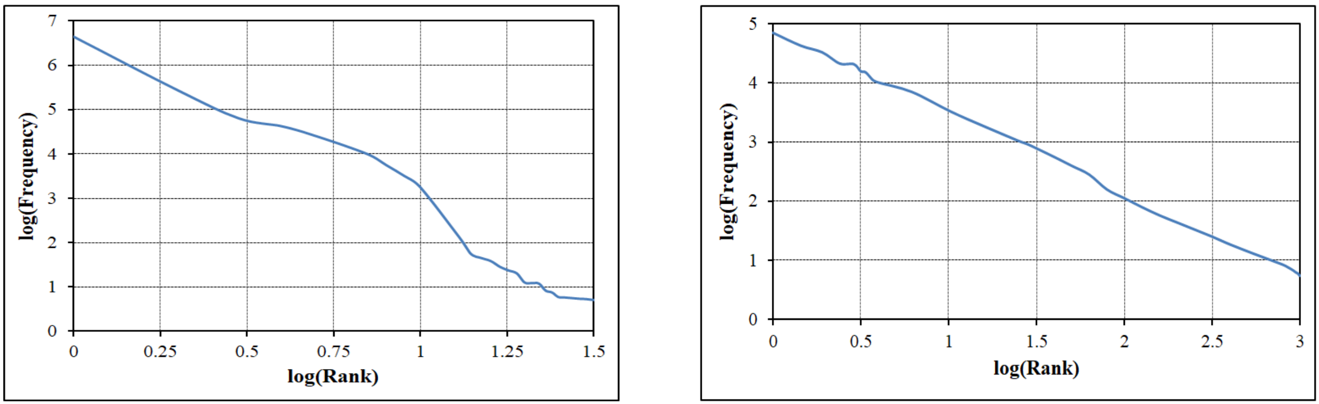


(a) Results in Bengali corpus

(b) Results in English corpus

Figure 4.11: Representativeness in our literature-based dataset according to Zipf's law

ure 4.11(b) respectively. Figure 4.11(a) shows that our Bengali corpus exhibits a bit deviation from Zipf's law; however, our English corpus perfectly follows Zipf's law. Similarly, we present the graphs for our second dataset (full dataset) in Figure 4.12(a) and Figure 4.12(b) respectively. Here, Figure 4.12(a) shows that our Bengali corpus of full dataset exhibits lesser deviation from Zipf's law than Bangali corpus of literature-based dataset due to combining literature-based dataset with custom dataset.



(a) Results in Bengali corpus

(b) Results in English corpus

Figure 4.12: Representativeness in our full dataset according to Zipf's law

4.3 Evaluation Metrics

Human evaluations of machine translation are extensive, however, expensive. Human evaluations can take substantial time to finish. Therefore, we need to adopt a quick method of automatic evaluation of machine translation, which can correlate highly with human evaluation. Accordingly, for the purpose of performance evaluation of our system, we adopt three different metrics that are widely used for evaluating performance of machine translation. The metrics are - 1) Bi-Lingual Evaluation Understudy (BLEU) [19], 2) Metric for Evaluation of Translation with Explicit ORdering (METEOR) [20], and 3) Translation Edit Rate (TER) [21]. We present brief overview on each of these metrics in the following subsections.

4.3.1 BLEU

BLEU presents an automated understudy to skilled human judges, which substitutes for them in case of a need for quick or frequent evaluations [19]. "The closer a machine translation is to a professional human translation, the better it is" - this is the theme of this method. Typically, there can be many "perfect" translations of a given source sentence. These translations may vary in word choice or in word order even when they use the same words. Yet, humans can clearly distinguish a good translation from a bad one. For example, let us consider two candidate translations of a source sentence in Example 1.

Example 1.

- **Candidate 1:** "It is a guide to action, which ensures that the military always obeys the commands of the party."
- **Candidate 2:** "It is to insure the troops forever hearing the activity guidebook that party direct."

Although they appear to convey the same meaning, they differ markedly in quality. For comparison, we state three reference human translations of the same sentence below.

- **Reference 1:** "It is a guide to action that ensures that the military will forever heed Party commands."
- **Reference 2:** "It is the guiding principle, which guarantees the military forces always being under the command of the Party."

- **Reference 3:** "It is the practical guide for the army always to heed the directions of the party."

It is clear that the good translation, Candidate 1, shares many words and phrases among these three reference translations, while Candidate 2 does not. Besides, note that Candidate 1 shares "It is a guide to action" with Reference 1, "which" with Reference 2, "ensures that the military" with Reference 1, "always" with References 2 and 3, "commands" with Reference 1, and finally "of the party" with Reference 2 (all ignoring capitalization). In contrast, Candidate 2 exhibits far fewer matches, and their extent is less.

It is clear that an automated program can rank Candidate 1 higher than Candidate 2 simply by comparing n-gram[1] matches between each candidate translation and the reference translations. Here, BLEU compares n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches are position independent. The more the matches, the better the candidate translation is.

To calculate BLEU score, we need to calculate the modified n-gram precision for the entire test corpus initially. To do so, first, we count the maximum number of times a candidate n-gram occurs in any single reference translation. Note that we compute these n-gram matches sentence by sentence. Next, we clip the total count of each candidate n-gram by its maximum reference count . In other words, we truncate each n-gram's count, if necessary, to not exceed the largest count observed in any single reference for that n-gram. Let us consider another example in this regard.

Example 2.

- **Candidate:** "the the the the the the the"
- **Reference 1:** "the cat is on the mat"
- **Reference 2:** "there is a cat on the mat"

In the example above, the unigram (n=1) "the" appears twice in reference 1, and once in reference 2. Thus, maximum reference count for the unigram is 2, whereas its total count in the candidate sentence is 7. Therefore, we clip its total count (7) by its maximum reference count (2).

Then, we sum up these clipped counts over all distinct n-grams for all the candidate sentences, and divide the summation by the total (unclipped) number of candidate n-grams in the test corpus.

[1] An n-gram is a contiguous sequence of n items from a given dataset. The items can be phonemes, syllables, letters, words, etc., according to the application. For example, if the sample sentence is - "This is an example", corresponding 1-grams (unigrams) are - "This", "is", "an", and "example", corresponding 2-grams (bigrams) are - "This is", "is an", and "an example", and so on.

Therefore, we can calculate the modified n-gram precision score, P_n for the entire test corpus as follows:

$$P_n = \frac{\sum_{C \in Candidates} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in Candidates} \sum_{n\text{-}gram \in C'} Count(n\text{-}gram')} \tag{4.1}$$

In Example 1, if we consider Candidate 1 is the only candidate sentence in the entire corpus then Candidate 1 (corpus) achieves a modified unigram precision of 17/18[2]. Similarly, Candidate 2 achieves a modified unigram precision of 8/14. Similarly, the modified unigram precision in Example 2 is 2/7. Besides, Candidate 1 achieves a modified bigram (n=2) precision of 10/17, whereas the lower quality Candidate 2 achieves a modified bigram precision of 1/13. In Example 2, the candidate sentence achieves a modified bigram precision of 0.

We penalize candidate translations longer than their references using the modified n-gram precision measure. Here, we introduce a multiplicative brevity penalty factor so that a high-scoring candidate translation must now match the reference translations in length, in word choice, and in word order. We also calculate the brevity penalty over the entire corpus to allow freedom at the sentence level. To do so, first, we compute the test corpus' effective reference length, r, by summing the best match lengths for each candidate sentence in the corpus. Next, we choose the brevity penalty through a decaying exponential as r/c, where c is the total length of the candidate translation corpus. We can calculate the brevity penalty (BP) as follows [19]:

$$BP = \begin{cases} 1, & \text{if } c > r. \\ e^{(1-r/c)}, & \text{otherwise.} \end{cases} \tag{4.2}$$

Finally, we calculate the BLEU score for the entire test corpus using the following formulas [19]:

$$BLEU = BP \times exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4.3}$$

$$\log BLEU = min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^{N} w_n \log p_n \tag{4.4}$$

[2] In Candidate 1, there are sixteen distinct unigrams - "It", "is", "a", "guide", "to", "action,", "which", "ensures", "that", "the", "military", "always", "obeys", "commands", "of", "party". Here, "the" has clipped count 3, "obeys" has 0, and each of the other unigrams has clipped count 1 contributing to total clipped count 17. Besides, total number of candidate unigrams is 18. Therefore, modified unigram precision is 17/18.

Here, we consider w_n's as the positive weights summing to one. In the baseline, we have chosen N=4 and uniform weights as w_n=1/N.

4.3.2 METEOR

METEOR is an automatic metric for machine translation evaluation that is based on a generalized concept of unigram matching between machine produced candidate translation and human produced reference translations [20]. METEOR can match unigrams based on their surface forms, stemmed forms, and meanings. Furthermore, we can easily extend METEOR to include more advanced matching strategies. Once Meteor finds all generalized unigram matches between the two strings, it computes a score for this matching using a combination of unigram-precision[3], unigram-recall[4], and a measure of fragmentation that is designed to directly capture how well-ordered the matched words in the machine translation are with respect to the reference.

METEOR evaluates a translation by computing a score based on explicit word-to-word matches between the translation and a reference translation. If more than one reference translation is available, METEOR scores the given translation against each reference independently, and reports the best score.

Given a pair of translations (a candidate sentence and a reference sentence) to be compared, METEOR creates an alignment between the two strings (translations). We define an alignment as a mapping between unigrams, such that every unigram in each string maps to zero or one unigram in the other string, and to no unigram in the same string. Thus, in a given alignment, a single unigram in one string cannot map to more than one unigram in the other string. We incrementally produce this alignment through a series of stages, each stage consisting of two distinct phases.

In the first phase, we list all the possible unigram mappings between the two strings. Thus, for example, if the word“computer” occurs once in the system translation and twice in the reference translation, we list two possible unigram mappings - one mapping the occurrence of “computer” in the system translation to the first occurrence of “computer” in the reference translation, and another mapping it to the second occurrence. Here, different modules map unigrams based on different criteria. The “exact” module maps two unigrams if they are exactly the same (e.g. “computers” maps to “computers” but not “computer”). The “porter stem” module maps two unigrams if they are the

[3] Unigram-precision is calculated as a ratio between the number of unigrams in the candidate translation that are also found in the reference translation and the total number of unigrams in the candidate translation.

[4] Unigram-recall is calculated as a ratio between the number of unigrams in the candidate translation that are also found in the reference translation and the total number of unigrams in the reference translation.

same after they are stemmed using the Porter stemmer (e.g.: "computers" maps to both "computers" and to "computer"). The "WN synonymy" module maps two unigrams if they are synonyms of each other (e.g.: "well" maps to "good").

In the second phase of each stage, we select the largest subset of these unigram mappings such that the resulting set constitutes an alignment as defined above (that is, each unigram must map to at most one unigram in the other string). If more than one subset constitutes an alignment, and also has the same cardinality as the largest set then we select the set that has the least number of unigram mapping crosses as shown in Figure 4.13.

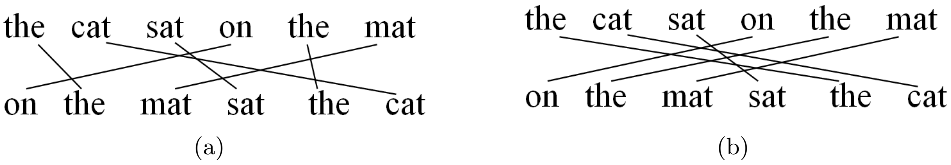


Figure 4.13: Unigram mappings between a candidate sentence and a reference sentence

Here, we choose the unigram mapping of Figure 4.13(a) over that of Figure 4.13(b) as Figure 4.13(a) has the least number of unigram mapping crosses. Formally, two unigram mappings (t_i, r_j) and (t_k, r_l) (where t_i and t_k are unigrams in the system translation mapped to unigrams r_j and r_l in the reference translation respectively) are said to cross if and only if the following formula evaluates to a negative number:

$$((pos(t_i) - pos(t_k)) \times (pos(r_j) - pos(r_l))) \tag{4.5}$$

Here, pos(t_x) is the numeric position of the unigram t_x in the system translation string, and pos(r_y) is the numeric position of the unigram r_y in the reference string.

Each stage only maps unigrams that we have not mapped to any unigram in any of the preceding stages. Generally, the first stage uses the "exact" mapping module, the second the "porter stem" module and the third the "WN synonymy" module. Once we have run all the stages and produced a final alignment between the candidate translation and the reference translation, we compute the METEOR score for this pair of sentences as follows. Firstly, we compute unigram precision (P) as the ratio of the number of unigrams in the system translation that are mapped (to unigrams in the reference translation) to the total number of unigrams in the system translation. Similarly, we compute unigram recall (R) as the ratio of the number of unigrams in the candidate sentence that are mapped (to unigrams in the reference sentence) to the total number of unigrams in the reference translation. Let us the consider a candidate sentence and a reference translation in the following example.

Example 3.

- **Candidate:** on the mat sat the cat
- **Reference:** the cat sat on the mat

In the above example, the number of mapped unigrams in the candidate sentence is 6, and the total number of unigrams in the candidate sentence is 6. Therefore, unigram precision, P is 1 (6/6). Besides, the total number of unigrams in the reference sentence is 6. Therefore, unigram recall, R is also 1 (6/6). Next, we compute F_{mean} by combining the precision and recall via a harmonic-mean that places most of the weight on recall. The resulting formula used is -

$$F_{mean} = \frac{10PR}{R + 9P} \tag{4.6}$$

To take into account longer matches, we calculate a penalty for a given alignment using the following formula:

$$Penalty = 0.5 \times \left(\frac{\text{\# of chunks}}{\text{\# of unigrams\_matched}} \right)^3 \tag{4.7}$$

For example, if the candidate sentence is "the president spoke to the audience" and the reference sentence is "the president then spoke to the audience", there are two chunks - "the president" and "spoke to the audience". Similarly, in our example (Example 3), there are six chunks (no bigram or longer matches) - "on", "the", "mat", "sat", "the", and "cat". We need to note that the penalty increases as the number of chunks increases to a maximum of 0.5. As the number of chunks goes to 1, penalty decreases. Finally, we compute the METEOR score for the chosen alignment as follows:

$$Score = F_{mean} \times (1 - Penalty) \tag{4.8}$$

This has the effect of reducing the F_{mean} by 50% maximum ($Penalty_{max} = 0.5$) if there are no bigram or longer matches. For example, we calculate the METEOR score for Example 3 as follows:

$$F_{mean} = \frac{10 \times 1 \times 1}{1 + 9 \times 1} = 1.00$$

$$Penalty = 0.5 \times \left(\frac{6}{6}\right)^3 = 0.50$$

$$Score = 1.00 \times (1 - 0.50) = 0.50$$

4.3.3 TER

Translation Edit Rate (TER) measures the amount of editing that a human would have to perform to change a system output so that it exactly matches a reference translation [21]. Formally, we define TER as the minimum number of edits (normalized by the average length of the references) needed to change a candidate sentence so that it exactly matches one of the references. Since the main concern is the minimum number of edits needed to modify the candidate, we measure only the number of edits to the closest reference. Specifically, we can calculate TER as follows:

$$TER = \frac{\text{Number of edits}}{\text{Average number of reference words}} \tag{4.9}$$

Possible edits include insertion, deletion, and substitution of single words as well as shifts of word sequences. A shift moves a contiguous sequence of words within the candidate sentence to another location within that sentence. All edits, including shifts of any number of words by any distance, have equal cost [21]. In addition to that, we treat punctuation tokens as normal words and count miscapitalization as an edit. For example, let us consider the reference-candidate pair below, where we indicate the differences between reference and candidate by upper case.

- **Reference:** SAUDI ARABIA denied THIS WEEK information published in the AMERICAN new york times.
- **Candidate:** THIS WEEK THE SAUDIS denied information published in the new york times.

Here, the candidate sentence is fluent and means the same thing (except for missing “American”) as the reference sentence. However, TER does not consider this an exact match. First, the phrase “this week” in the candidate is in a shifted position (at the beginning of the sentence rather than after the word “denied”) with respect to the reference. Second, the phrase “Saudi Arabia” in the reference appears as “the Saudis” in the candidate (this counts as two separate substitutions). Finally, the word “American” appears only in the reference. If we apply TER to this candidate and reference, the number of edits is 4 (1 shift, 2 substitutions, and 1 insertion), giving a TER score of 4/13=31%.

We calculate the number of edits for TER in two phases. In the first phase, we use a greedy search[5] to find the set of shifts, by repeatedly selecting the shift that most reduces the number of insertions, deletions, and substitutions, until no more beneficial shift remains. In the next phase, we use dynamic programming to optimally calculate the remaining edit distance using a minimum-edit-distance (where each insertion, deletion, or substitution has a cost of 1) [21]. We calculate the number of edits for all of the references, and take the best (lowest) score.

The greedy search is necessary to select the set of shifts because an optimal sequence of edits (with shifts) is very expensive to find. We use several other constraints in order to further reduce the space of possible shifts and to allow for efficient computation. These constraints are intended to simulate the way in which a human editor might choose the words to shift. They are as follows:

1. The shifted words must exactly match the reference words in the destination position.

2. The word sequence of the candidate in the original position and the corresponding reference words must not exactly match. This prevents the shifting of words that are currently correctly matched.

3. The word sequence of the reference that corresponds to the destination position must be misaligned before the shift. This prevents shifting to align the words that already correctly aligned.

As an example, let us consider the following reference-hypothesis pair:

Reference:	a	b	c	d	e	f		c	
Candidate:	a			d	e		b	c	f

Here, we can shift the words “b c” in the candidate to the left to correspond to the words “b c” in the reference, because there is a mismatch in the current location of “b c” in the candidate, and there is a mismatch of “b c” in the reference. After the shift the candidate changes to as follows:

Reference:	a	b	c	d	e	f	c
Candidate:	a	b	c	d	e	f	

TER, as defined above, only calculates the number of edits between the best reference and the candidate. If we use TER in the case of multiple references, it most accurately measures the error rate of a candidate sentence when the corresponding reference is the closest possible reference to the candidate.

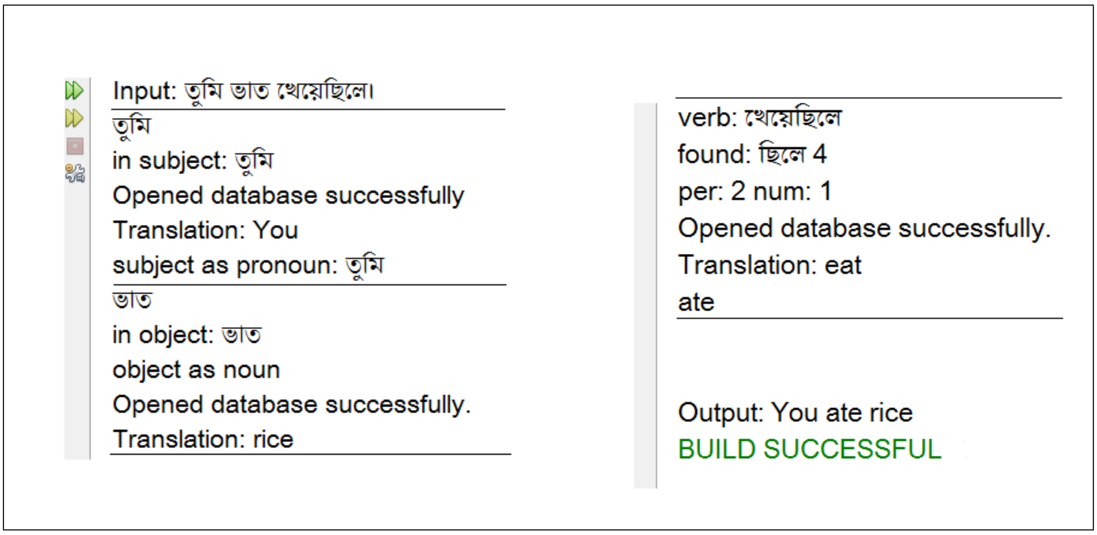
[5] Since the solution to this is conjectured to be NP-hard, a greedy search is used here [21].

4.4 Experimental Results and Findings

Based on the above-mentioned performance metrics, we evaluate performances of our proposed techniques through rigorous experimentation. We present results and finding of the evaluation in the next subsections.

4.4.1 Results from Our Proposed Rule-based Translator

In our rule-based translator, we consider all types of sentences covering basic simple, complex, and compound sentences. Initially, we implement simple sentences in our system through adding some basic rules for forming a simple sentence. Figure 4.14 and Figure 4.15 show a couple of glimpses of sample outputs from our implementation for translating simple sentences using JAVA.

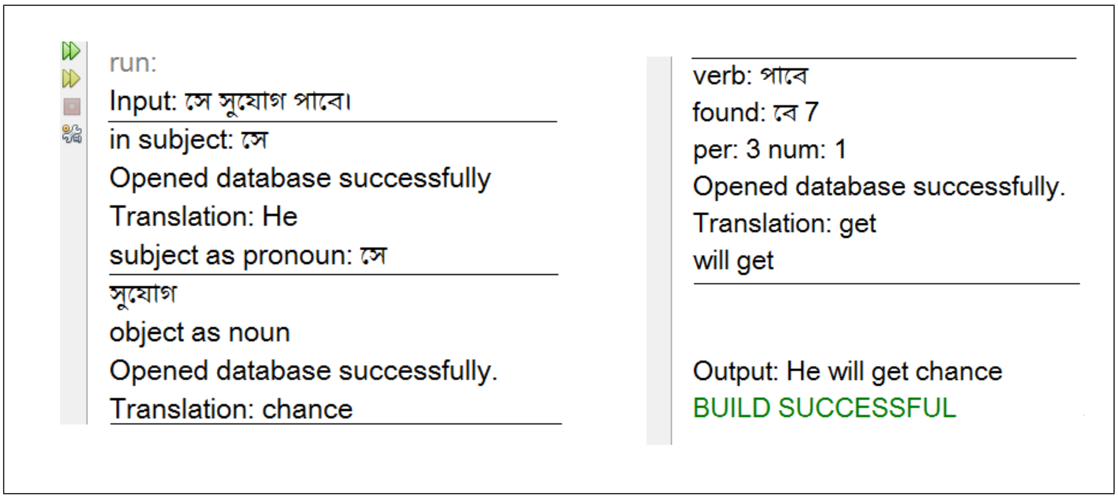


```
Input: তুমি ভাত খেয়েছিলে।
তুমি
in subject: তুমি
Opened database successfully
Translation: You
subject as pronoun: তুমি
ভাত
in object: ভাত
object as noun
Opened database successfully.
Translation: rice
```

```
verb: খেয়েছিলে
found: ছিলে 4
per: 2 num: 1
Opened database successfully.
Translation: eat
ate


Output: You ate rice
BUILD SUCCESSFUL
```

Figure 4.14: Sample translation of simple sentences (simple past tense)

In Figure 4.14, we show translation of a sentence in simple past tense. Here, our rule-based translation system analyze the input Bengali sentence and synthesizes it. Here, first, our system determines the subject and recognizes the subject as a pronoun. Then, our system identifies the object as a noun. Next, our system identifies the verb and analyzes carefully to properly detect the tense. Besides, our system also recognizes person and number of the subject. Moreover, analyzing the suffix of the verb, our system detects the tense. Apart from this, our system can generate translations of all the words (subject, object, and verb) in the input sentence from its vocabulary as shown in the



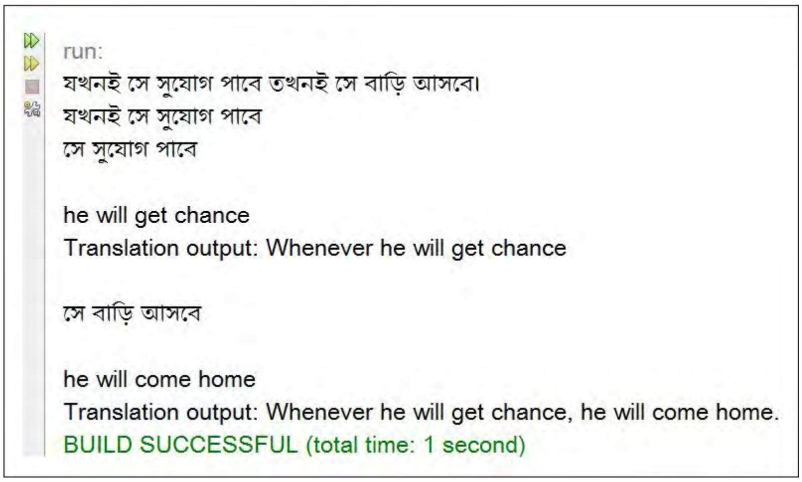
```
run:
Input: সে সুযোগ পাবে।
in subject: সে
Opened database successfully
Translation: He
subject as pronoun: সে
সুযোগ
object as noun
Opened database successfully.
Translation: chance
```

```
verb: পাবে
found: বে 7
per: 3 num: 1
Opened database successfully.
Translation: get
will get


Output: He will get chance
BUILD SUCCESSFUL
```

Figure 4.15: Sample translation of simple sentences (simple future tense)

figure. Later, it modifies the verb by adding suffixes according to the tense. This is how our system generates the final output (translated sentence). Similarly, Figure 4.15 shows another example of translating a sentence in simple future tense.

Now, we focus on a complex sentence in Figure 4.16. Here, first, our system recognizes the complex sentence by examining the presence of any keyword of complex sentence in the input sentence. Next, our system splits the complex sentence into two independent simple sentences. Then, our system

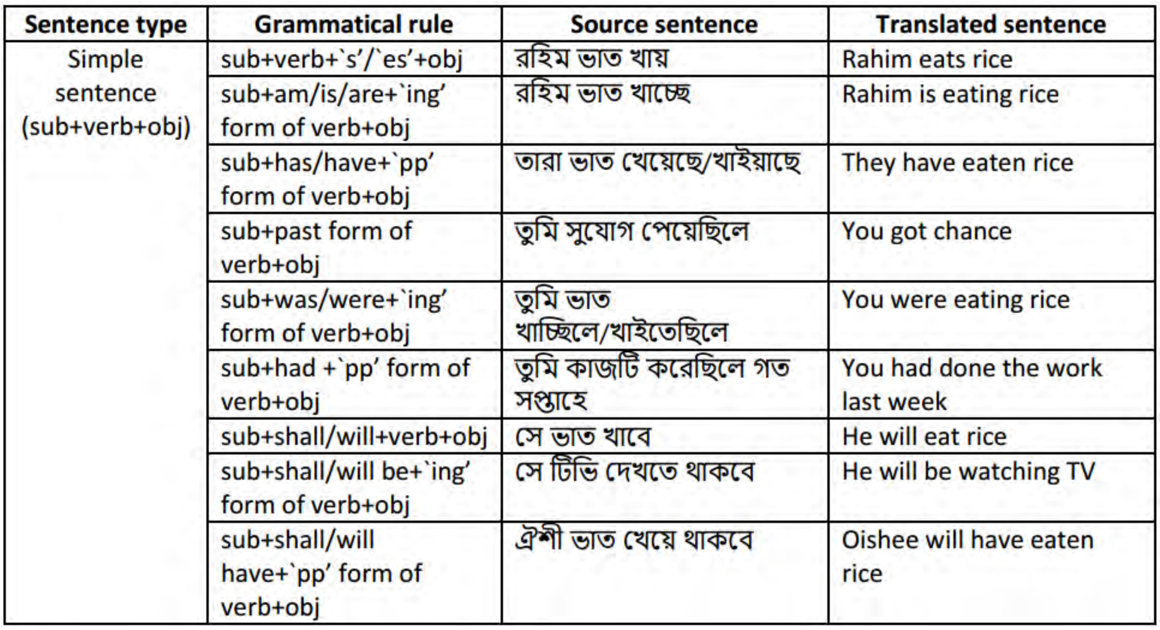


```
run:
যখনই সে সুযোগ পাবে তখনই সে বাড়ি আসবে।
যখনই সে সুযোগ পাবে
সে সুযোগ পাবে

he will get chance
Translation output: Whenever he will get chance

সে বাড়ি আসবে

he will come home
Translation output: Whenever he will get chance, he will come home.
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 4.16: Sample translation of a complex sentence

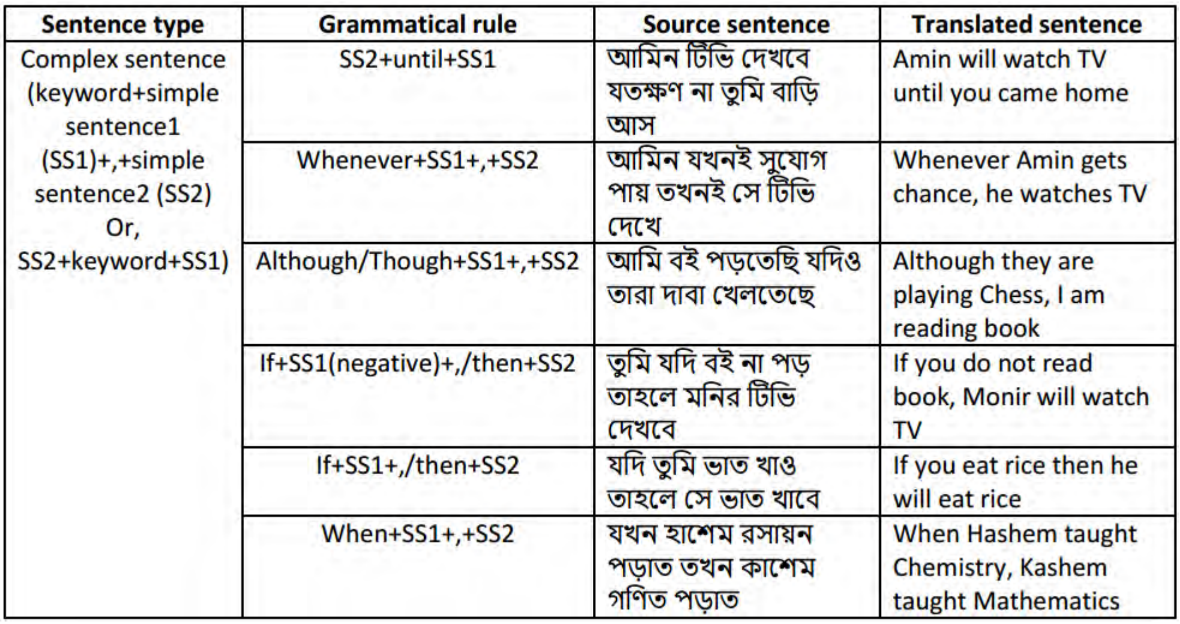
translates these simple sentences using the same procedure used for translating simple sentences as discussed earlier. Finally, our system combines the translated simple sentences with necessary keywords to generate the target translated sentence.

We test our rule-based translator with different types of sentences (from our datasets), which realize a number of different rules. We show a scenario of our experimental results in Table 4.2, Table 4.3, and Table 4.4. Here, we summarize some of our implemented rules and generated outputs. Table 4.2 shows a partial list of our implemented rules used for translating simple sentences. Table 4.3 and Table 4.4 show some of our implemented rules and some examples of generated translations for complex and compound sentences respectively.



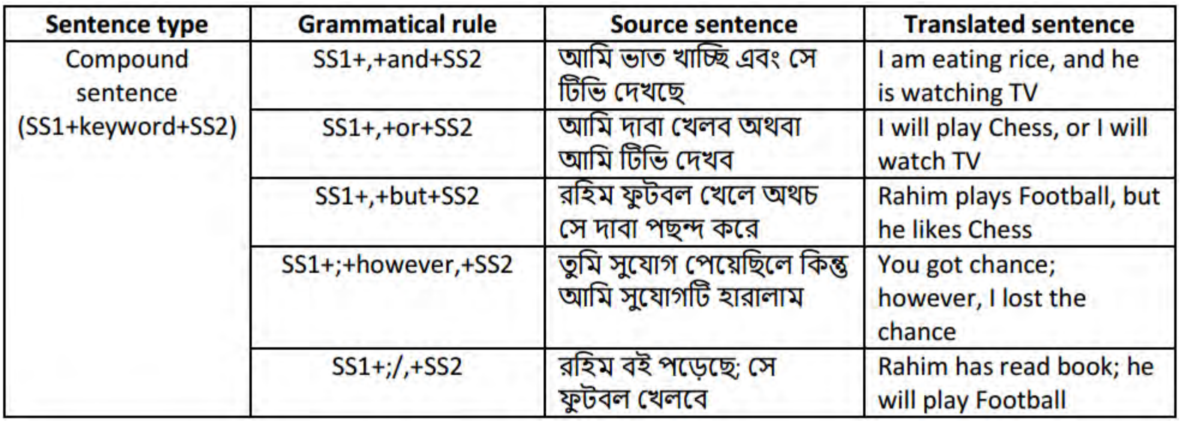
Sentence type	Grammatical rule	Source sentence	Translated sentence
Simple sentence (sub+verb+obj)	sub+verb+`s'/`es'+obj	রহিম ভাত খায়	Rahim eats rice
	sub+am/is/are+`ing' form of verb+obj	রহিম ভাত খাচ্ছে	Rahim is eating rice
	sub+has/have+`pp' form of verb+obj	তারা ভাত খেয়েছে/খাইয়াছে	They have eaten rice
	sub+past form of verb+obj	তুমি সুযোগ পেয়েছিলে	You got chance
	sub+was/were+`ing' form of verb+obj	তুমি ভাত খাচ্ছিলে/খাইতেছিলে	You were eating rice
	sub+had +`pp' form of verb+obj	তুমি কাজটি করেছিলে গত সপ্তাহে	You had done the work last week
	sub+shall/will+verb+obj	সে ভাত খাবে	He will eat rice
	sub+shall/will be+`ing' form of verb+obj	সে টিভি দেখতে থাকবে	He will be watching TV
	sub+shall/will have+`pp' form of verb+obj	ঐশী ভাত খেয়ে থাকবে	Oishee will have eaten rice

Table 4.2: Experimental results for some example simple sentences



Sentence type	Grammatical rule	Source sentence	Translated sentence
Complex sentence (keyword+simple sentence1 (SS1)+,+simple sentence2 (SS2) Or, SS2+keyword+SS1)	SS2+until+SS1	আমিন টিভি দেখবে যতক্ষণ না তুমি বাড়ি আস	Amin will watch TV until you came home
	Whenever+SS1+,+SS2	আমিন যখনই সুযোগ পায় তখনই সে টিভি দেখে	Whenever Amin gets chance, he watches TV
	Although/Though+SS1+,+SS2	আমি বই পড়তেছি যদিও তারা দাবা খেলতেছে	Although they are playing Chess, I am reading book
	If+SS1(negative)+,/then+SS2	তুমি যদি বই না পড় তাহলে মনির টিভি দেখবে	If you do not read book, Monir will watch TV
	If+SS1+,/then+SS2	যদি তুমি ভাত খাও তাহলে সে ভাত খাবে	If you eat rice then he will eat rice
	When+SS1+,+SS2	যখন হাশেম রসায়ন পড়াত তখন কাশেম গণিত পড়াত	When Hashem taught Chemistry, Kashem taught Mathematics

Table 4.3: Experimental results for some example complex sentences

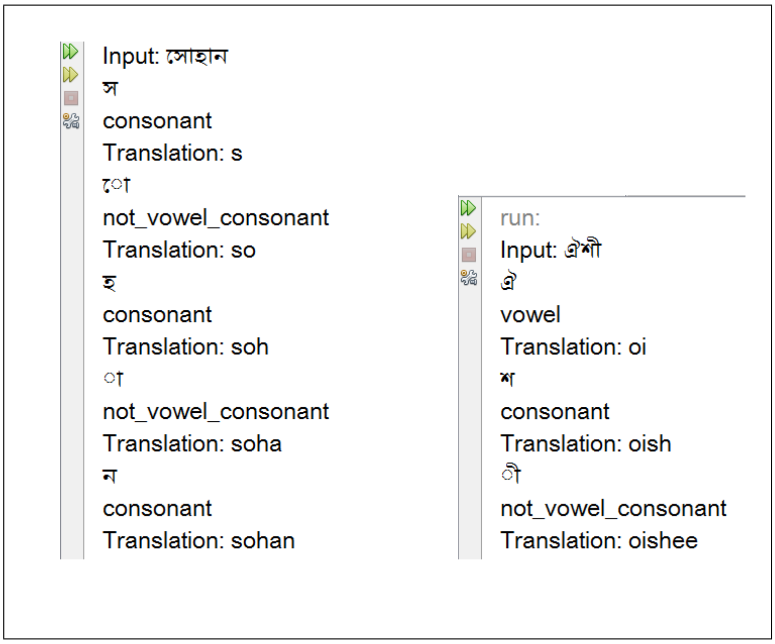


Sentence type	Grammatical rule	Source sentence	Translated sentence
Compound sentence (SS1+keyword+SS2)	SS1+,+and+SS2	আমি ভাত খাচ্ছি এবং সে টিভি দেখছে	I am eating rice, and he is watching TV
	SS1+,+or+SS2	আমি দাবা খেলব অথবা আমি টিভি দেখব	I will play Chess, or I will watch TV
	SS1+,+but+SS2	রহিম ফুটবল খেলে অথচ সে দাবা পছন্দ করে	Rahim plays Football, but he likes Chess
	SS1+;+however,+SS2	তুমি সুযোগ পেয়েছিলে কিন্তু আমি সুযোগটি হারালাম	You got chance; however, I lost the chance
	SS1+;/,+SS2	রহিম বই পড়েছে; সে ফুটবল খেলবে	Rahim has read book; he will play Football

Table 4.4: Experimental results for some example compound sentences

4.4.2 Results on Name Identification

Next, we present outcomes of our name identification mechanism and corresponding translations. As discussed earlier, one of the major improvements by our rule-based translator is name identification and name translation.

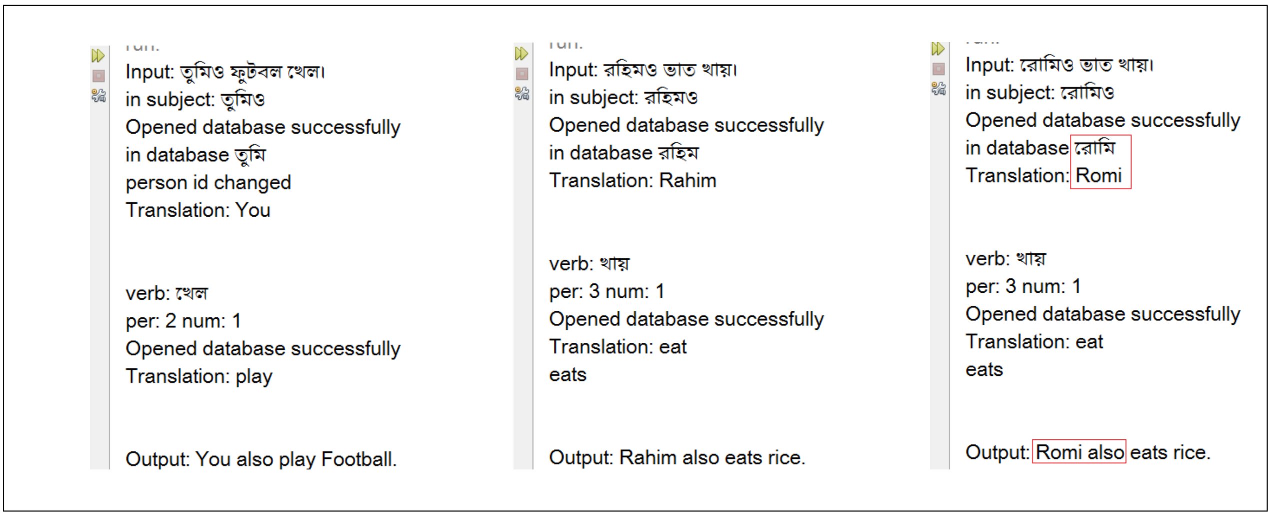


```
Input: সোহান
স
consonant
Translation: s
ো
not_vowel_consonant
Translation: so
হ
consonant
Translation: soh
া
not_vowel_consonant
Translation: soha
ন
consonant
Translation: sohan
```

```
run:
Input: ঐশী
ঐ
vowel
Translation: oi
শ
consonant
Translation: oish
ী
not_vowel_consonant
Translation: oishee
```

Figure 4.17: Sample outputs of name identifications and translating names

Figure 4.17 shows translation of two names - ‘Sohan’ and ‘Oishee’, generated by our system. Besides, our system can also process subjects with emphasizing tags after separating the emphasizing tags as shown in Figure 4.18. However, one important observation regarding emphasized subject identification is that the emphasizing tag itself may be the part of a valid name (subject) as mentioned earlier in the previous chapter (Chapter 3). In such (less frequent) cases, our system discards that tag (which is not actually any tag) from the valid name leading to a faulty name identification. In Figure 4.18, our system misinterprets ‘Romio’ by removing the tag and thus reducing it to ‘Romi’. Note that such cases are quite rare and less contributing to performance compared to the overall improvement achieved in most of the cases. Therefore, we tolerate this shortcoming in our system as a trade-off, and leave its solution as a future work.



```
run:
Input: তুমিও ফুটবল খেল।
in subject: তুমিও
Opened database successfully
in database তুমি
person id changed
Translation: You


verb: খেল
per: 2 num: 1
Opened database successfully
Translation: play


Output: You also play Football.
```

```
run:
Input: রহিমও ভাত খায়।
in subject: রহিমও
Opened database successfully
in database রহিম
Translation: Rahim


verb: খায়
per: 3 num: 1
Opened database successfully
Translation: eat
eats


Output: Rahim also eats rice.
```

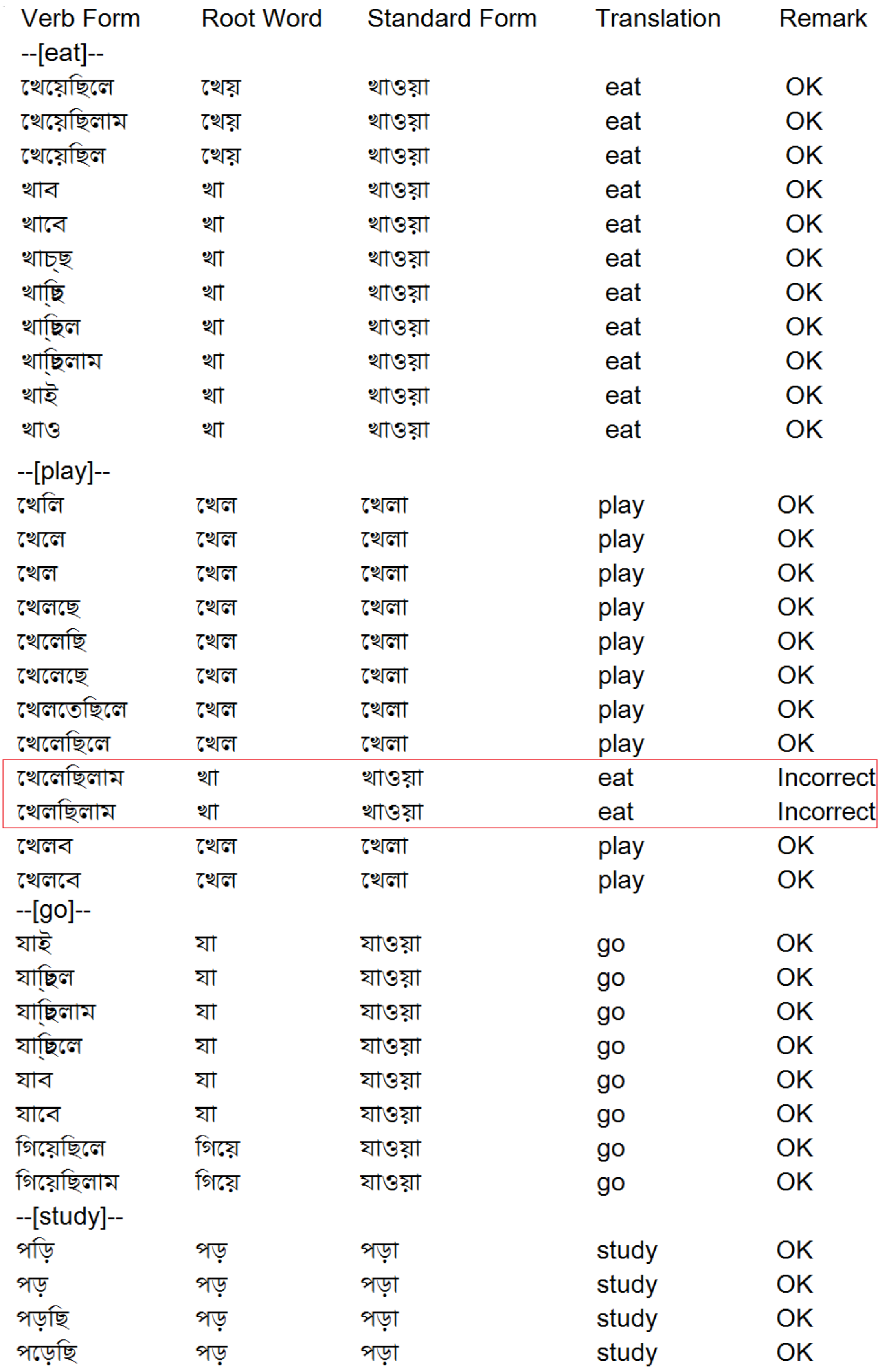
```
run:
Input: রোমিও ভাত খায়।
in subject: রোমিও
Opened database successfully
in database রোমি
Translation: Romi


verb: খায়
per: 3 num: 1
Opened database successfully
Translation: eat
eats


Output: Romi also eats rice.
```

Figure 4.18: Processing of subjects with emphasizing tags

4.4.3 Results on Optimized Verb Translation Technique

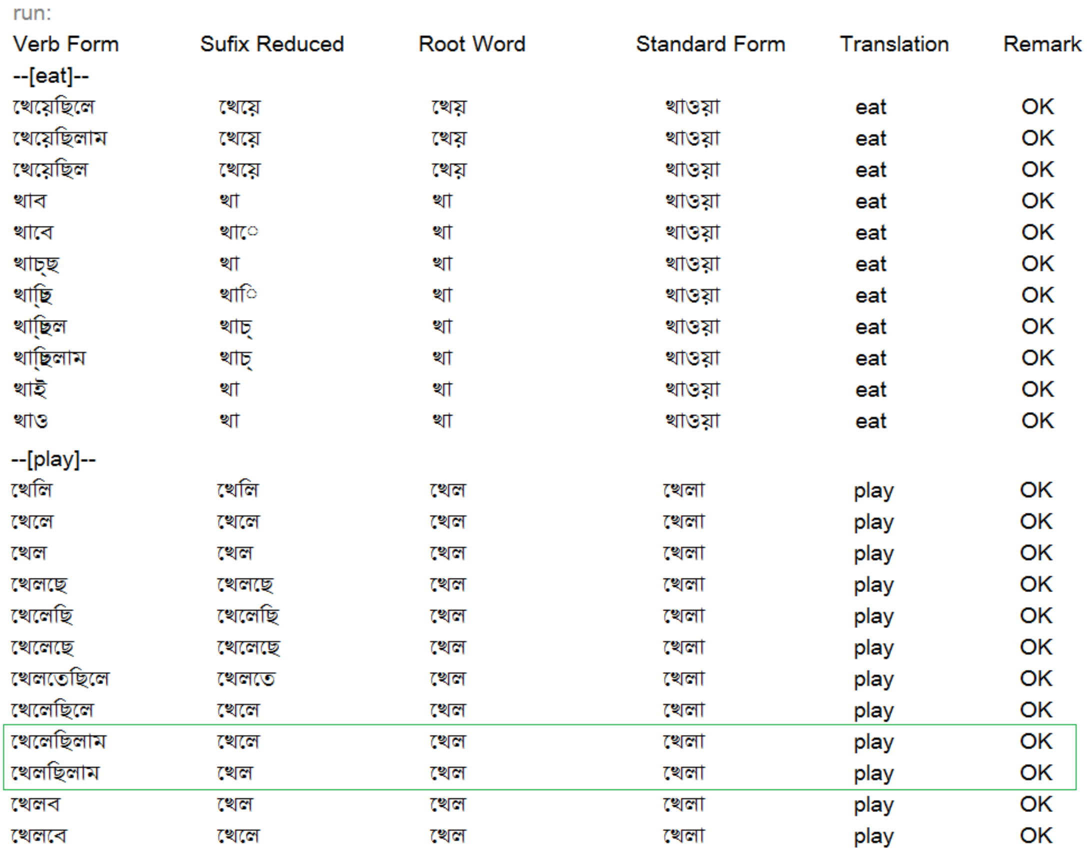
In case of our optimized verb identification technique, modified Levenshtein distance calculation shows significant improvement in terms of optimizing both memory consumption and searching time. We apply this algorithm on several Bengali verbs to get their root verbs, which we then map to standard forms of the verbs. We present outcomes of our modified Levenshtein distance algorithm in Figure 4.19.

Note that, detection of root verb by calculating Levenshtein distance can be incorrect for some forms of verbs, which can lead to incorrect translations of those verbs. In Figure 4.19, we can notice one such case where our first modification of the Levenshtein distance algorithm provides erroneously translated verb 'eat' in place of 'play'. We handle such erroneous cases successfully through preprocessing of the verbs before determining the root verbs as discussed earlier in Chapter 3. Figure 4.20 shows the improvement achieved (inside green box) after incorporating the preprocessing of verbs before calculating Levenshtein distance. Here, we overcome the incorrect detection of root verbs shown previously in Figure 4.19 and accomplish detection of correct root verbs for almost all the possible cases. Afterwards, our system translates the verb by modifying its raw translated form as per other relevant information (POS tagging, person, number, etc.) extracted from the input sentence.



Verb Form	Root Word	Standard Form	Translation	Remark
--[eat]--				
খেয়েছিলে	খেয়	খাওয়া	eat	OK
খেয়েছিলাম	খেয়	খাওয়া	eat	OK
খেয়েছিল	খেয়	খাওয়া	eat	OK
খাব	খা	খাওয়া	eat	OK
খাবে	খা	খাওয়া	eat	OK
খাচ্ছ	খা	খাওয়া	eat	OK
খাচ্ছি	খা	খাওয়া	eat	OK
খাচ্ছিল	খা	খাওয়া	eat	OK
খাচ্ছিলাম	খা	খাওয়া	eat	OK
খাই	খা	খাওয়া	eat	OK
খাও	খা	খাওয়া	eat	OK
--[play]--				
খেলি	খেল	খেলা	play	OK
খেলে	খেল	খেলা	play	OK
খেল	খেল	খেলা	play	OK
খেলছে	খেল	খেলা	play	OK
খেলেছি	খেল	খেলা	play	OK
খেলেছে	খেল	খেলা	play	OK
খেলতেছিলে	খেল	খেলা	play	OK
খেলেছিলে	খেল	খেলা	play	OK
খেলেছিলাম	খা	খাওয়া	eat	Incorrect
খেলছিলাম	খা	খাওয়া	eat	Incorrect
খেলব	খেল	খেলা	play	OK
খেলবে	খেল	খেলা	play	OK
--[go]--				
যাই	যা	যাওয়া	go	OK
যাচ্ছিল	যা	যাওয়া	go	OK
যাচ্ছিলাম	যা	যাওয়া	go	OK
যাচ্ছিলে	যা	যাওয়া	go	OK
যাব	যা	যাওয়া	go	OK
যাবে	যা	যাওয়া	go	OK
গিয়েছিলে	গিয়ে	যাওয়া	go	OK
গিয়েছিলাম	গিয়ে	যাওয়া	go	OK
--[study]--				
পড়ি	পড়	পড়া	study	OK
পড়	পড়	পড়া	study	OK
পড়ছি	পড়	পড়া	study	OK
পড়েছি	পড়	পড়া	study	OK

Figure 4.19: Outcomes of our first modification on Levenshtein distance algorithm



```
run:
Verb Form        Sufix Reduced    Root Word        Standard Form    Translation    Remark
--[eat]--
খেয়েছিলে          খেয়ে             খেয়              খাওয়া             eat            OK
খেয়েছিলাম         খেয়ে             খেয়              খাওয়া             eat            OK
খেয়েছিল           খেয়ে             খেয়              খাওয়া             eat            OK
খাব               খা               খা               খাওয়া             eat            OK
খাবে              খাে              খা               খাওয়া             eat            OK
খাচ্ছ              খা               খা               খাওয়া             eat            OK
খাচ্ছি             খাি              খা               খাওয়া             eat            OK
খাচ্ছিল            খাচ্              খা               খাওয়া             eat            OK
খাচ্ছিলাম          খাচ্              খা               খাওয়া             eat            OK
খাই               খা               খা               খাওয়া             eat            OK
খাও               খা               খা               খাওয়া             eat            OK

--[play]--
খেলি              খেলি              খেল              খেলা              play           OK
খেলে              খেলে              খেল              খেলা              play           OK
খেল               খেল               খেল              খেলা              play           OK
খেলছে             খেলছে             খেল              খেলা              play           OK
খেলেছি            খেলেছি            খেল              খেলা              play           OK
খেলেছে            খেলেছে            খেল              খেলা              play           OK
খেলতেছিলে         খেলতে             খেল              খেলা              play           OK
খেলেছিলে          খেলে              খেল              খেলা              play           OK
খেলেছিলাম         খেলে              খেল              খেলা              play           OK
খেলছিলাম          খেল               খেল              খেলা              play           OK
খেলব              খেল               খেল              খেলা              play           OK
খেলবে             খেলে              খেল              খেলা              play           OK
```

Figure 4.20: Further improvement over modified Levenshtein distance through removing common suffixes

4.4.4 Overall Improvement with Name Identification and Optimized Verb Translation Technique

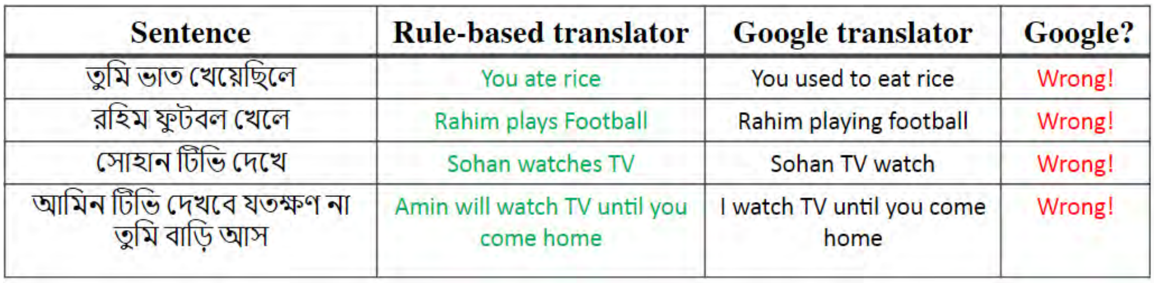
We present the improvement achieved by our rule-based translator after implementing name identification and optimized verb translation technique in terms of BLEU score in Table 4.5. Table 4.5 shows the improvement achieved using our both individual sentences (540 sentences) and full dataset (11,500 sentences). Here, we achieve significant improvement using our individual sentences as these sentences are designed based on the set of rules implemented in our rule-based translator. As discussed earlier (in Chapter 3), we extract individual sentences mainly for testing the performance of our rule-based translator so that they remain in-line with the different rules implemented in our rule-based translator.

Dataset	**Rule-based**	**Improved Rule-based**	**Improvement**
Individual sentences	71.28	92.36	30%
Full	3.05	3.13	3%

Table 4.5: Improvement with name identification and optimized verb translation technique in terms of BLEU score

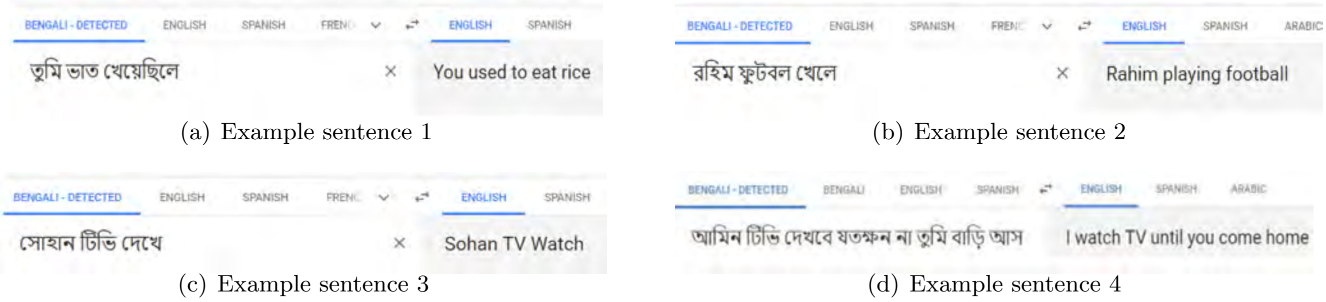
4.4.5 Comparison with Google Translator

We compare the performance of our rule-based translator with that of popular Google Translator. In the comparison, we present that our rule-based translator performs better than Google Translator in case of sentences whose rules have already been implemented in our system so far. We show examples of such improvements achieved by our rule-based translator over Google Translator in Table 4.6.



Sentence	Rule-based translator	Google translator	Google?
তুমি ভাত খেয়েছিলে	You ate rice	You used to eat rice	Wrong!
রহিম ফুটবল খেলে	Rahim plays Football	Rahim playing football	Wrong!
সোহান টিভি দেখে	Sohan watches TV	Sohan TV watch	Wrong!
আমিন টিভি দেখবে যতক্ষণ না তুমি বাড়ি আস	Amin will watch TV until you come home	I watch TV until you come home	Wrong!

Table 4.6: Comparison between performances of our rule-based translator and Google Translator for some example sentences



তুমি ভাত খেয়েছিলে

You used to eat rice

(a) Example sentence 1

রহিম ফুটবল খেলে

Rahim playing football

(b) Example sentence 2

সোহান টিভি দেখে

Sohan TV Watch

(c) Example sentence 3

আমিন টিভি দেখবে যতক্ষন না তুমি বাড়ি আস

I watch TV until you come home

(d) Example sentence 4

Figure 4.21: Snapshots of translations generated by Google Translator for our example sentences in Table 4.5 (collected on or before August 30, 2019)

We show all of these translations generated by Google Translator [45] for our example sentences in Figure 4.21.

4.4.6 Results from Our Different Blending Approaches

After implementation of both our proposed rule-based translator and classical NMT, we blend between these two approaches using three different techniques as discussed earlier. We analyze performances of each of these approaches with three standard metrics namely BLEU, METEOR, and TER as presented earlier. We consider different types of datasets with different sizes for analyzing the performances. This is because, results obtained from only one dataset may not be enough to draw any convincing conclusion about performance of translation by our proposed different approaches. Therefore, as already mentioned, we adopt a literature-based dataset (from Al-Quran) and create another dataset from different sources except any literature. The latter dataset, i.e., our custom dataset, is relatively smaller in size (around 3,500 parallel sentence pairs), which is too small to train an NMT system reasonably. Therefore, we combine this dataset with our literature-based dataset to form another dataset (full dataset) for experimentation.

4.4.6.1 Results using Literature-based Dataset

First, we present results (scores of performance metrics) obtained from translation over our literature-based dataset in Table 4.7.

Score	NMT	Rule-based	NMT+rule-based	Rule-based+NMT	NMT or rule-based
BLEU	8.56	1.28	11.43	0.84	8.80
METEOR	12.34	13.50	20.31	10.62	12.43
TER	93.73	93.90	85.09	96.62	93.50

Table 4.7: Comparison among different translation approaches

Table 4.7 shows a comparison among all the approaches (in isolation and in combination) using the standard performance metrics. Here, the higher the METEOR score and the BLEU score, and the lower the TER score; the better the performance is. From Table 4.7, we notice that ‘NMT followed by rule-based’ (NMT+rule-based) blending technique exhibits significant improvement over the classical NMT (GNMT). More specifically, it emerges as the best blending technique that gets reflected in the performance scores using each of the three metric. Therefore, we can understand that our blending approaches can significantly improve performance of NMT generated translations. The best way of blending appears to be applying grammatical rules after translating by NMT.

Another blending technique, ‘either NMT or rule-based’ (NMT or rule-based), also shows slight improvement over the classical NMT. We actually anticipate that since we carefully choose the best between the translations from rule-based translator and NMT as per the types of sentences in this technique. However, performance scores decline in ‘rule-based followed by NMT’ (rule-based+NMT) blending approach, which points out the inability of NMT to further improve translations done by the rule-based translation. Table 4.8 reflects a closer look at BLEU scores of all the approaches as per consideration of different n-grams (n=1, 2, 3, and 4). Ideally, BLEU score is considered for n-gram model where n=4. In all cases including n=4, the blending of ‘NMT followed by rule-based’ outperforms all other alternatives.

n-gram	**NMT**	**Rule-based**	**NMT+rule-based**	**Rule-based+NMT**	**NMT or rule-based**
1-gram	31.46	31.74	46.07	25.59	31.35
2-gram	17.18	10.70	25.19	7.56	17.37
3-gram	11.42	3.70	16.04	2.60	11.66
4-gram	8.56	1.28	11.43	0.84	8.80

Table 4.8: Comparison as per BLEU scores

Next, we present comparisons over classical NMT and other approaches graphically to portray the individual performance scores for each of the test sentences. We show comparisons using METEOR and TER scores where light red lines indicate NMT score and deep blue lines indicate each one of the other approaches one by one. Here, we adopt NMT as the benchmark approach in all the graphs, as it is commonly adopted by the widely-used Google translator. Note that we do not show any comparison in terms of BLEU score at sentence level as BLEU is generally calculated over the entire test corpus.

Comparison between NMT and Only Rule-based Approach

Firstly, Figure 4.22 and Figure 4.23 show a comparison between NMT and only rule-based approach (deep blue lines) in isolation in terms of METEOR and TER scores respectively. These two figures reflect that the overall performance of only rule-based approach is worse than NMT in isolation for literature-based dataset.

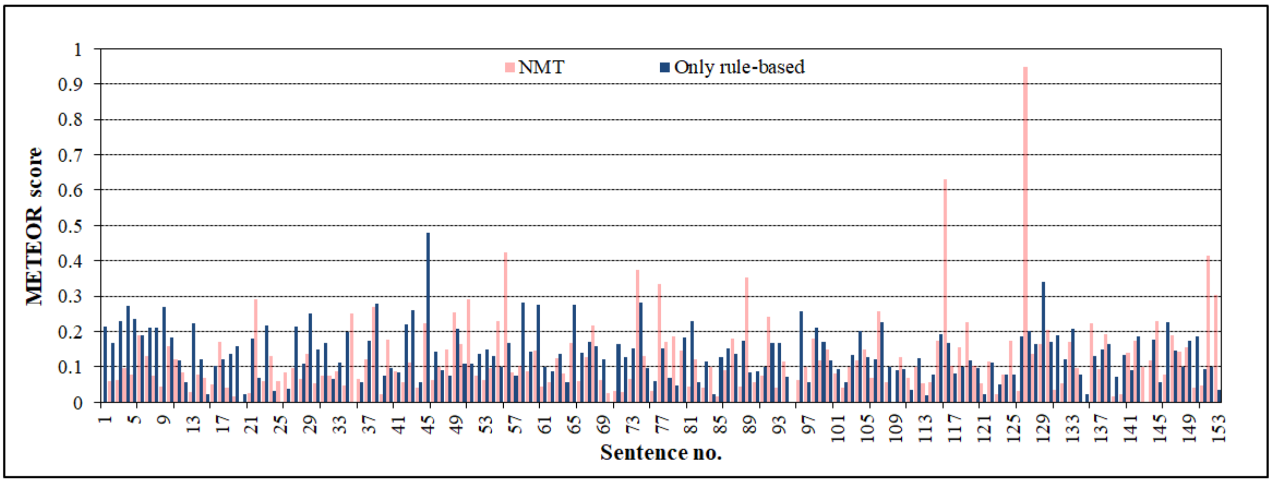


Figure 4.22: NMT versus only rule-based METEOR score

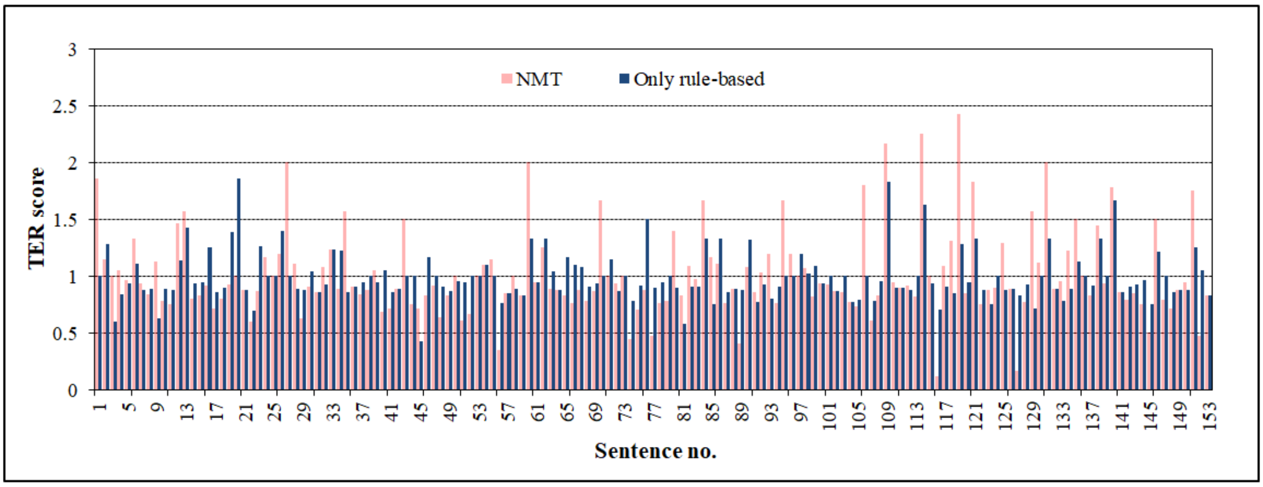


Figure 4.23: NMT versus only rule-based TER score

Comparison between NMT and 'NMT followed by Rule-based' Approach

Next, we show the performance of one of our blending techniques, 'NMT followed by rule-based' (NMT+rule-based), in terms of METEOR and TER scores in Figure 4.24 and Figure 4.25 respectively. Actually, these two figures reflect the performance of our best blending technique in terms of METEOR and TER scores. Here, deep blue lines indicate the scores obtained using 'NMT followed by rule-based' blending approach. We can see significant improvement over classical NMT in these figures.

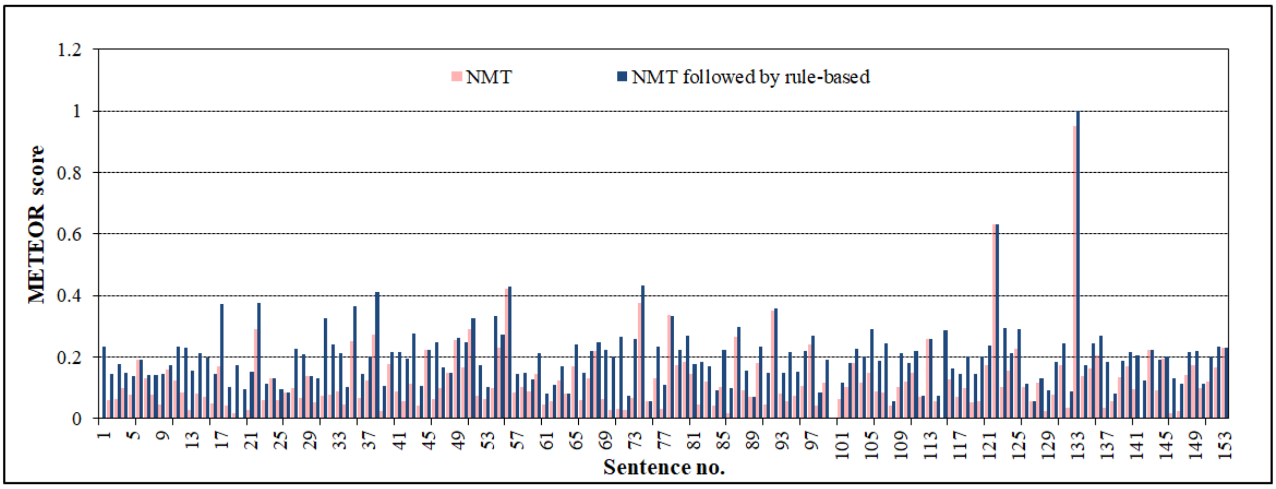


Figure 4.24: NMT versus NMT followed by rule-based METEOR score

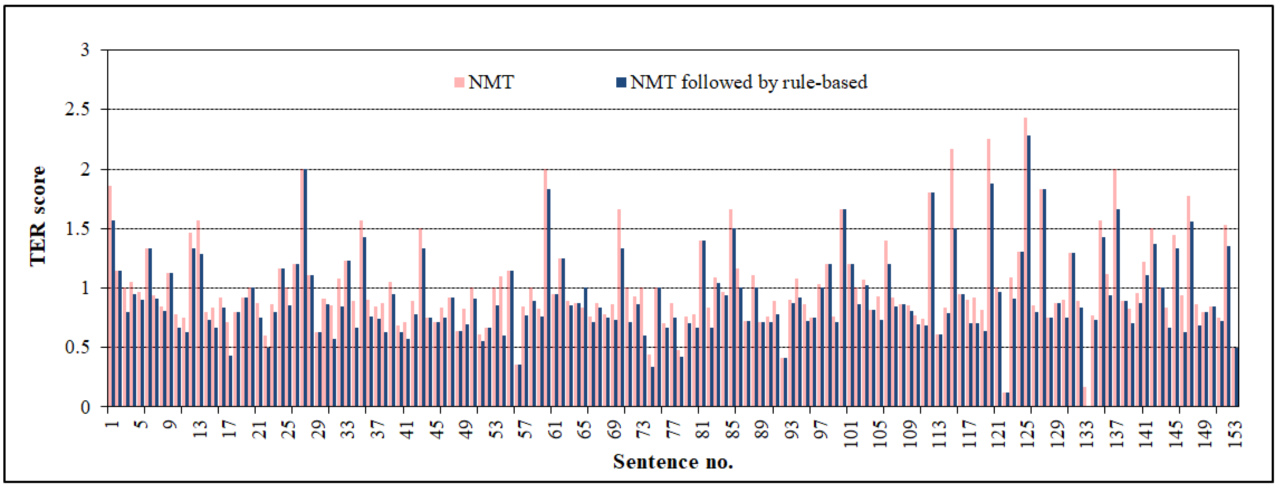


Figure 4.25: NMT versus NMT followed by rule-based TER score

Comparison between NMT and ‘Rule-based followed by NMT’ Approach

After that, we present the results of ‘rule-based followed by NMT’ (rule-based+NMT), in terms of METEOR and TER scores in Figure 4.26 and Figure 4.27 respectively. Here, deep blue lines indicate the scores obtained using ‘rule-based followed by NMT blending’ approach. These two figures reflect that ‘rule-based followed by NMT’ approach performs poorly when compared to the classical NMT. In fact, this blending technique proves itself to be the worst performer among all the approaches.

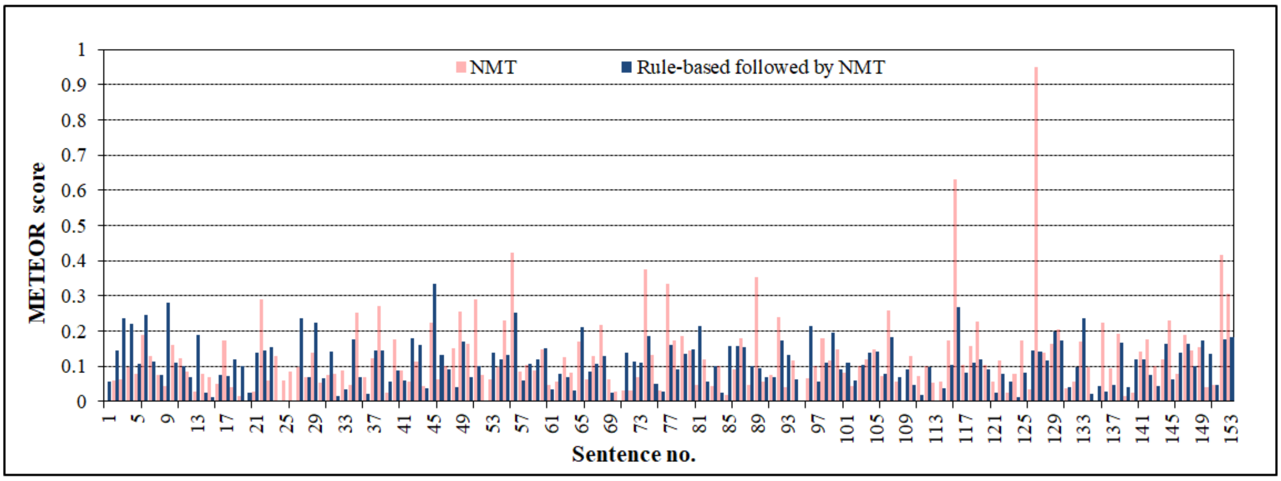


Figure 4.26: NMT versus rule-based followed by NMT METEOR score

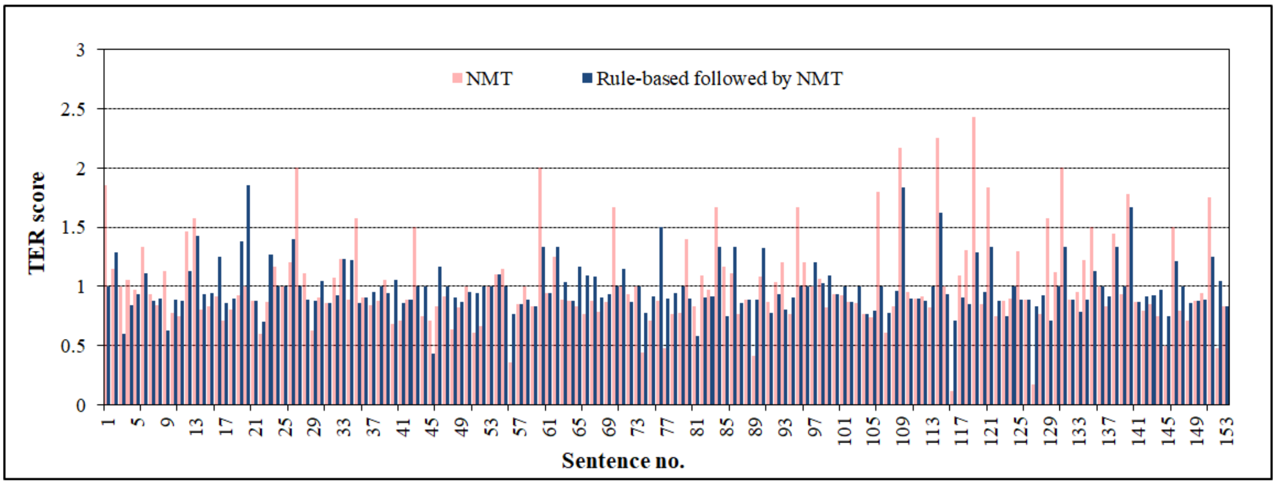


Figure 4.27: NMT versus rule-based followed by NMT TER score

Comparison between NMT and ‘Either NMT or Rule-based’ Approach

Finally, we present the results of ‘either NMT or rule-based’ blending technique in Figure 4.28 and Figure 4.29. Here, deep blue lines indicate the scores obtained using this blending approach. This approach performs on par with classical NMT as shown in the figures. Main reason behind this result is that most of the sentences are lengthy in this dataset. Since this approach chooses NMT generated translation if the length of the sentence is large, it chooses NMT generated translations mostly. However, this approach performs at least as good as classical NMT.

We can clearly notice that light red lines exceed deep blue lines for most of the sentences in

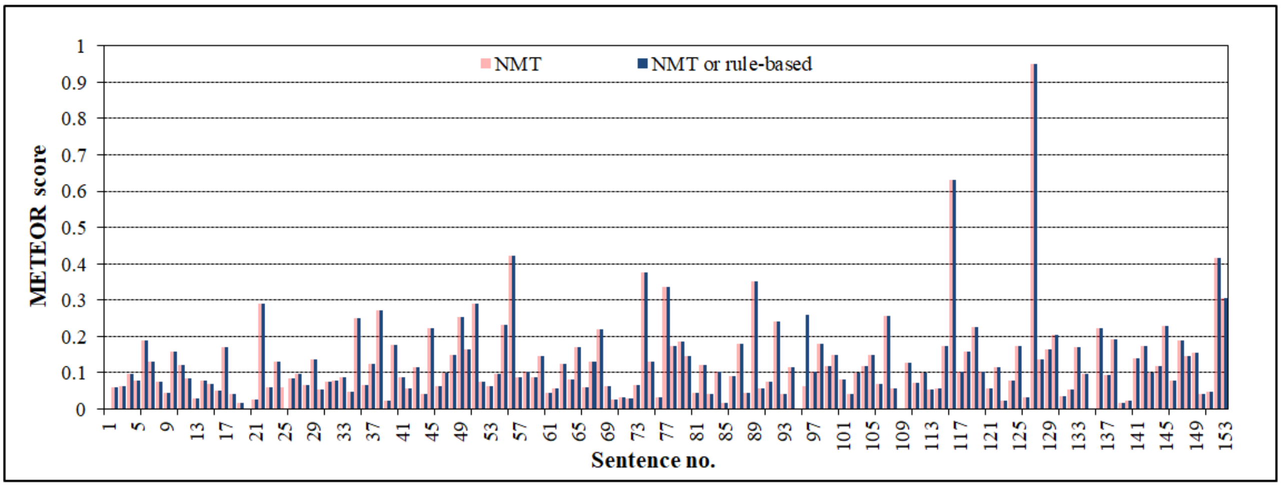


Figure 4.28: NMT versus NMT or rule-based METEOR score

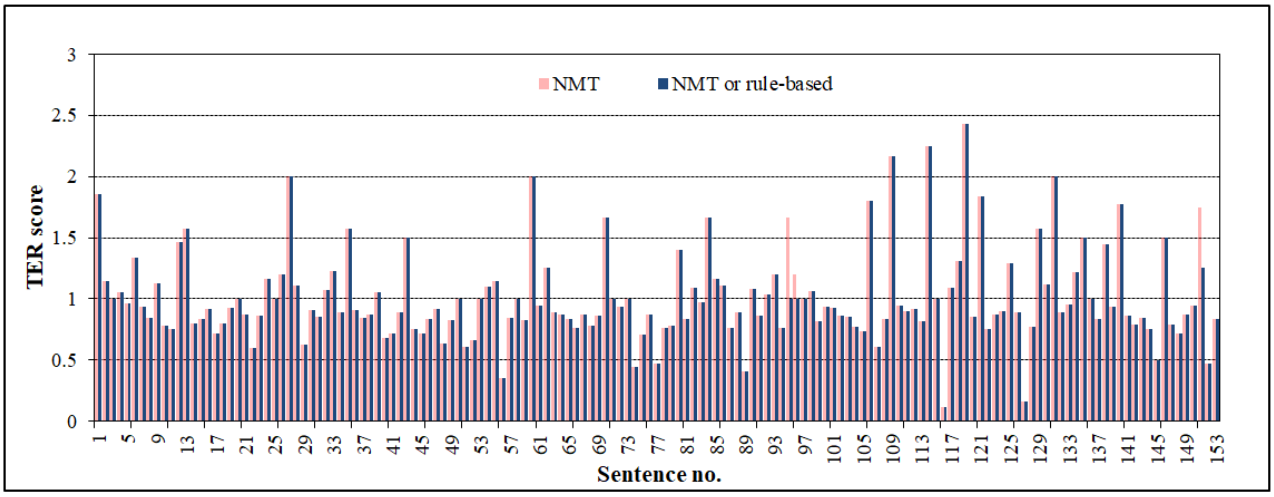


Figure 4.29: NMT versus NMT or rule-based TER score

Figure 4.22 and Figure 4.26. That means, both only rule-based approach and 'rule-based followed by NMT' approach perform worse than NMT in isolation. However, deep blue lines exceed light red lines in Figure 4.24 for almost all the sentences, which reflects the clear victory of our 'NMT followed by rule-based' approach over NMT in isolation. In addition to that, we notice that light red lines and deep blue lines are mostly at the same level in Figure 4.28, which reflects the on par performance of our 'either NMT or rule-based' approach as discussed above.

Analysis on Sensitivity of Our Operational Parameter

Performance of our rule-based translator changes as we increase the number of rules or we add more rules. However, adding rules seems like a never-ending process. Therefore, we analyze how implementation of different numbers of rules impacts on the performance scores of our different approaches.

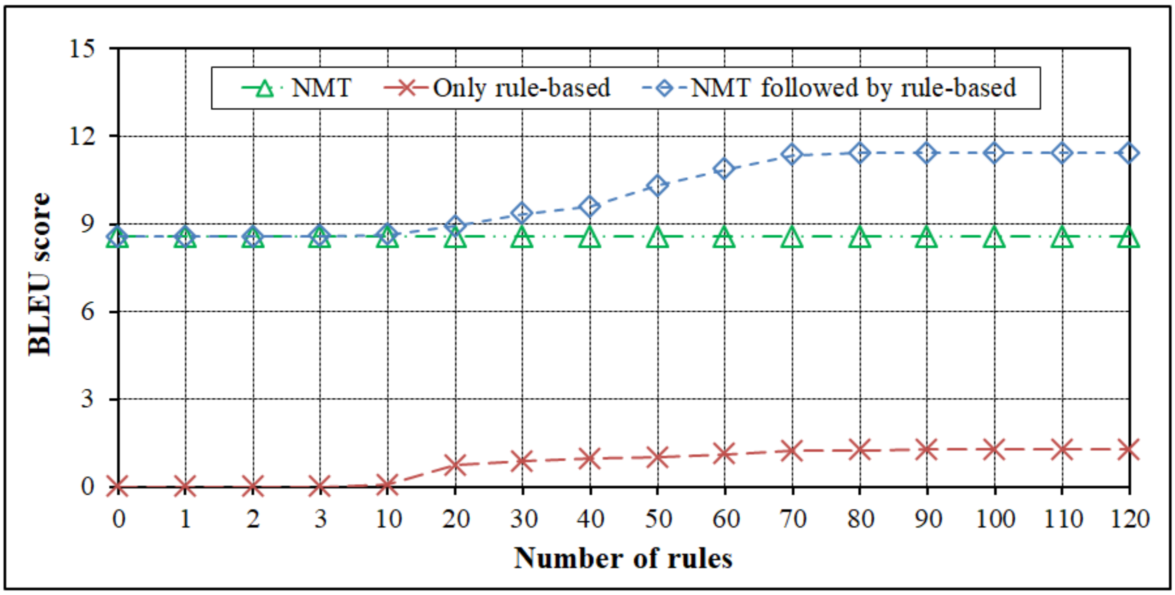


Figure 4.30: Variation of BLEU scores with an increase in the number of implemented rules

BLEU score increases as number of implemented rules increases in our system as shown in Figure 4.30. In this figure, we show performance of three different approaches with respect to an increase in the number of added rules - only rule-based approach, NMT, and 'NMT followed by rule-based' approach. We notice that the curves of only rule-based approach and 'NMT followed by rule-base' approach show a gradual increase (initially sharp) in BLEU score as the number of implemented rules increases. Besides, the curves tend to become flat after implementing around 90-100 rules in our system. It depends on the order in which different rules are being added. In our system, we implement more basic and important grammatical rules such as basic sentence structures (Table 4.2, Table 4.3, and Table 4.4), verb identification, tenses, etc., first. That is why, the curve shows a sharp rise in between first 3-10 implemented rules, and then rises consistently until 70-80 rules are added. In our system, we add the most important rules that significantly improve the translation performance within around first 50 rules (specifically, rule number 30-50). Afterwards, addition of more rules merely impacts on changing the performance score significantly since those rules such as detection of subject's gender, punctuations, etc., seem to be less contributing compared to the previously added (first 50-60 rules) rules.

Nonetheless, the curve for NMT remains flat (parallel to X axis) since performance of NMT does not change with the number of implemented rules. Moreover, although the curve of 'NMT followed by rule-based' approach exhibits characteristics nearly similar to that of only rule-based approach, it does not directly originate from the curve of rule-based approach using any mathematical formula. However, if the performance of translation generated by only rule-based approach improves then our blending ('NMT followed by rule-based' approach) also improves its performance to some extent since our system blends with that improved rule-based translation after generating translation by NMT. This is why, we notice such similarity between these two curves.

Similarly, we illustrate variation of METEOR scores with respect to the number of added rules. Figure 4.31 presents the results for only rule-based approach, NMT, and 'NMT followed by rule-based' approach.

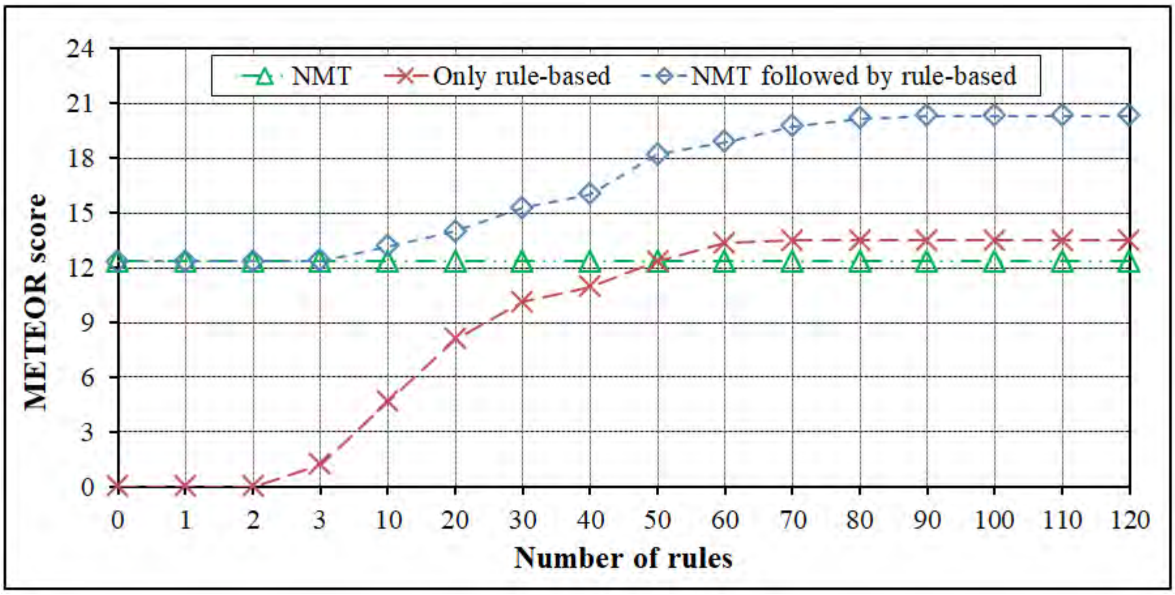


Figure 4.31: Variation of METEOR scores with an increase in the number of implemented rules

Trends in these curves for METEOR scores of these three approaches are similar to what we have just presented for BLEU scores above. Here, we show variation for only 'NMT followed by rule-based' approach, as this approach leads all other approaches. Note that the curves in this approach does not start from zero score, as NMT already sets a positive score that our blending system further increases by applying rules.

Finally, we also show variation of TER scores with an increase in the number of rules in Figure 4.32 for only rule-based approach, NMT, and 'NMT followed by rule-based' approach. Expectedly, apart from curve of NMT, behaviour of remaining two curves for TER scores is exactly opposite to the previous curves, as TER scores decrease with an increase in the number of rules. Here, less score refers to better performance, as the score refers to an error rate. Similar to the previous case, the curves go almost flat after the addition of first 70 rules.

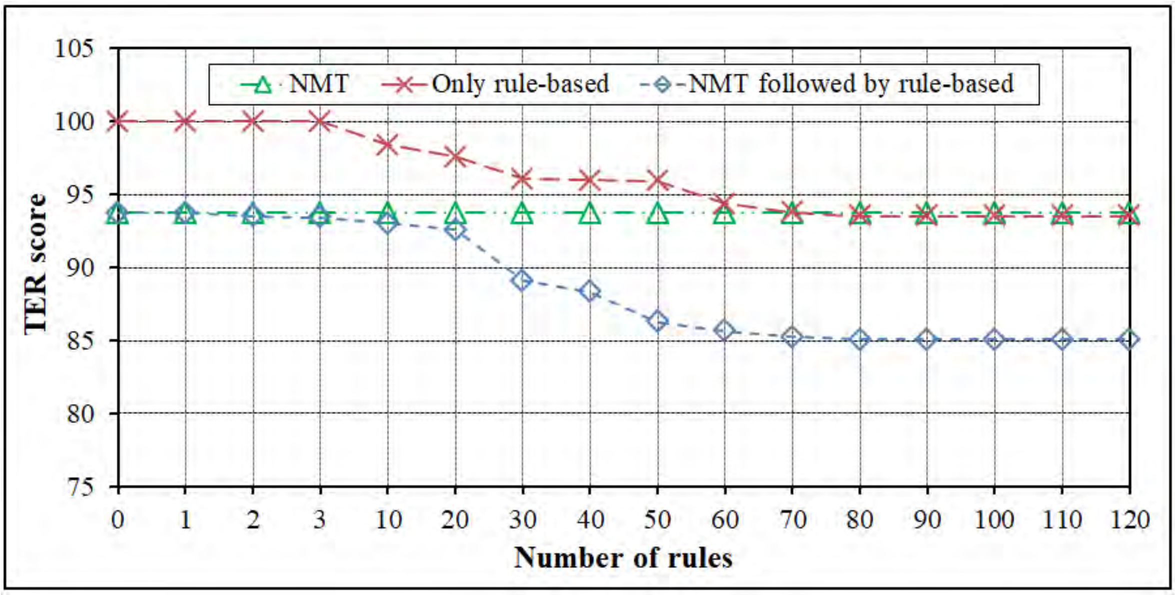


Figure 4.32: Variation of TER scores with an increase in the number of implemented rules

We present a combined graph (Figure 4.33) containing normalized value of all the metrics for both rule-based approach and 'NMT followed by rule-based' approach. Here, in case of values of each metric, we normalize the values with respect to our found maximum values. The combined presentation of all the normalized values in Figure 4.33 demonstrates efficacy of our proposed blending approach, as its application improves performance metrics in all cases.

That is all about experimentation on performance scores using our literature-based dataset. However, as promised earlier, we also perform similar experimentation using another dataset (full dataset) since scores obtained from only one dataset may not be enough to draw any convincing conclusion on

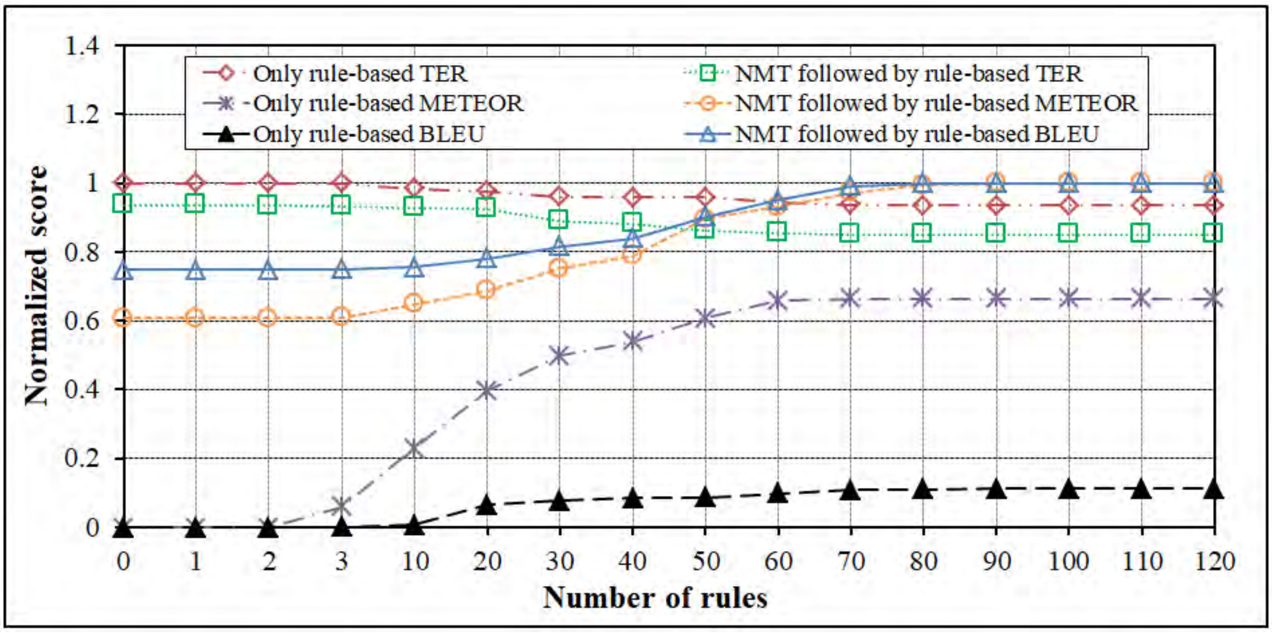


Figure 4.33: Comparison of normalized performance scores with an increase in the number of implemented rules for literature-based dataset

translation performance.

4.4.7 Results using Full Dataset

Next, we perform experimentation using our combined (literature-based and custom) dataset. Table 4.9 shows summary of results obtained using this dataset.

Score	NMT	Rule-based	NMT+rule-based	Rule-based+NMT	NMT or rule-based
BLEU	9.28	3.13	12.26	1.34	9.87
METEOR	14.18	14.43	22.32	12.86	14.92
TER	92.78	93.21	83.83	95.52	92

Table 4.9: Comparison among different translation approaches for full (combined) dataset

Table 4.9 strongly supports the results obtained earlier (Table 4.7) using our literature-based dataset. Here, ‘NMT followed by rule-based’ blending approach again outperforms all other approaches. In addition to that, ‘Either NMT or rule-based’ approach remains as our second best approach. Next, similar to our literature-based dataset, we show variation of BLEU, METEOR, and TER scores with an increase in the number of rules in Figure 4.34, Figure 4.35, and Figure 4.36

respectively for only rule-based approach, NMT, and 'NMT followed by rule-based' approach using our full dataset.

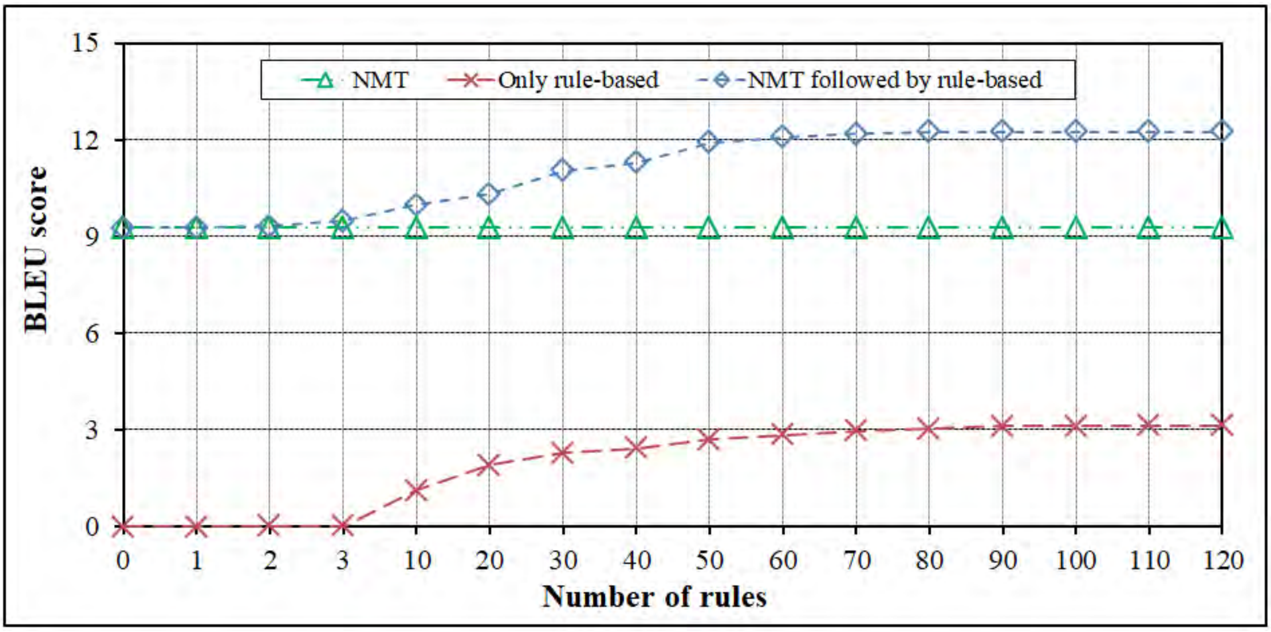


Figure 4.34: Variation of BLEU scores with an increase in the number of implemented rules for full dataset

Besides, we also present another combined graph (Figure 4.37) containing normalized value of all the metrics for both rule-based approach and 'NMT followed by rule-based' approach using our full dataset. Figure 4.37 reflects that the graph for our full dataset exhibits similar behaviour with respect to our previous dataset (literature-based). Therefore, we have just double-checked and justified our observation on performance scores of different approaches discussed earlier (for literature-based dataset), using our combined dataset this time.

Besides, we perform analysis on time and memory consumption for our implemented methods with different datasets. To do so, we first calculate average time and memory separately required by NMT and rule-based translator for translating a sentence. Then, we determine required blending time and memory while applying each of our blending techniques. We find that NMT requires around 80 minutes and 90 minutes for training with our literature-based dataset and full dataset respectively. After that, NMT performs inference for generating translations based on its learning acquired through training. We consider only this inference time (or translation time) in determining time required for translating each sentence by NMT. On an average, NMT requires 0.08-0.09s (80-90ms) for generating

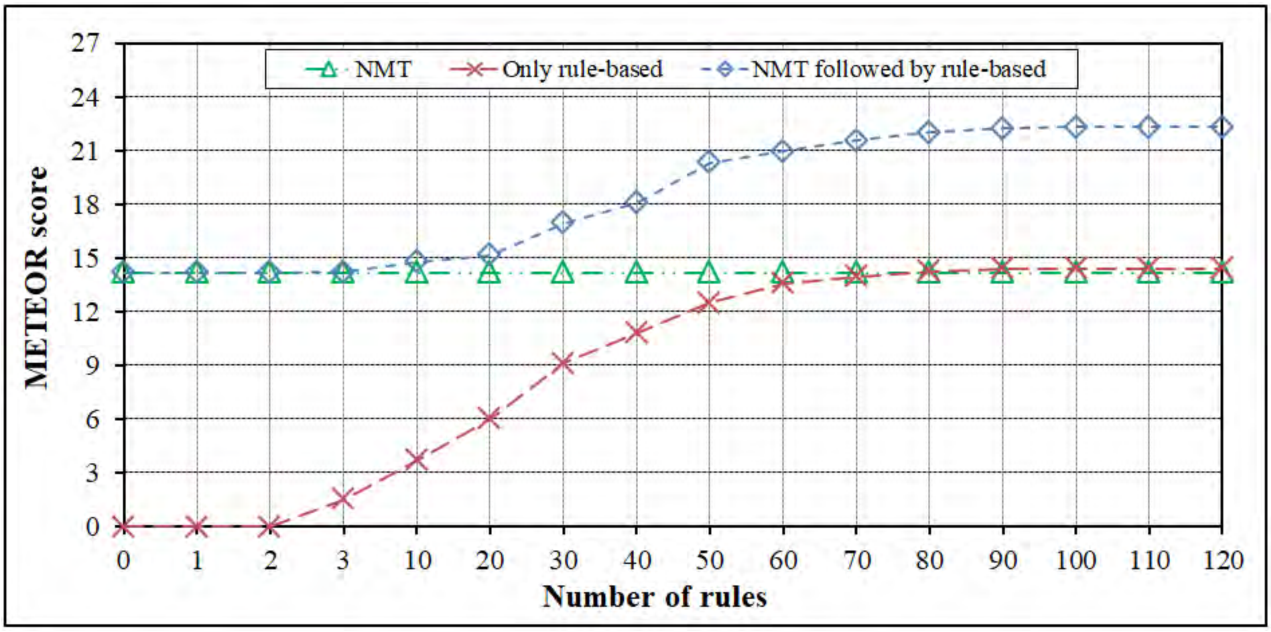


Figure 4.35: Variation of METEOR scores with an increase in the number of implemented rules for full dataset

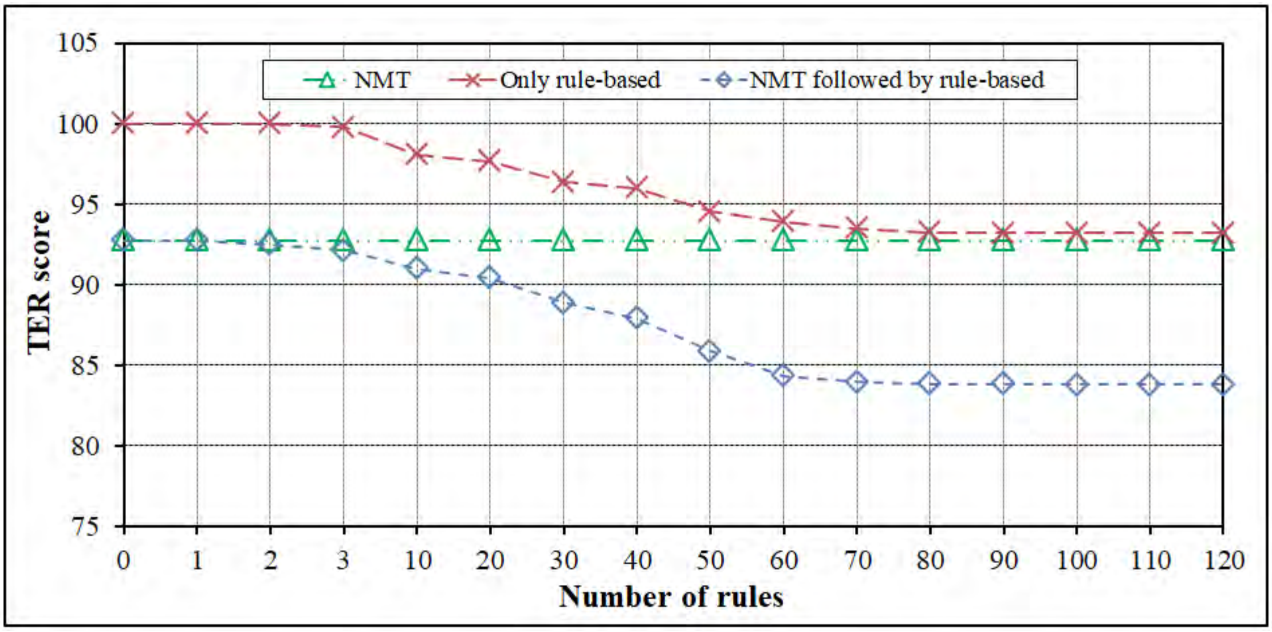


Figure 4.36: Variation of TER scores with an increase in the number of implemented rules for full dataset

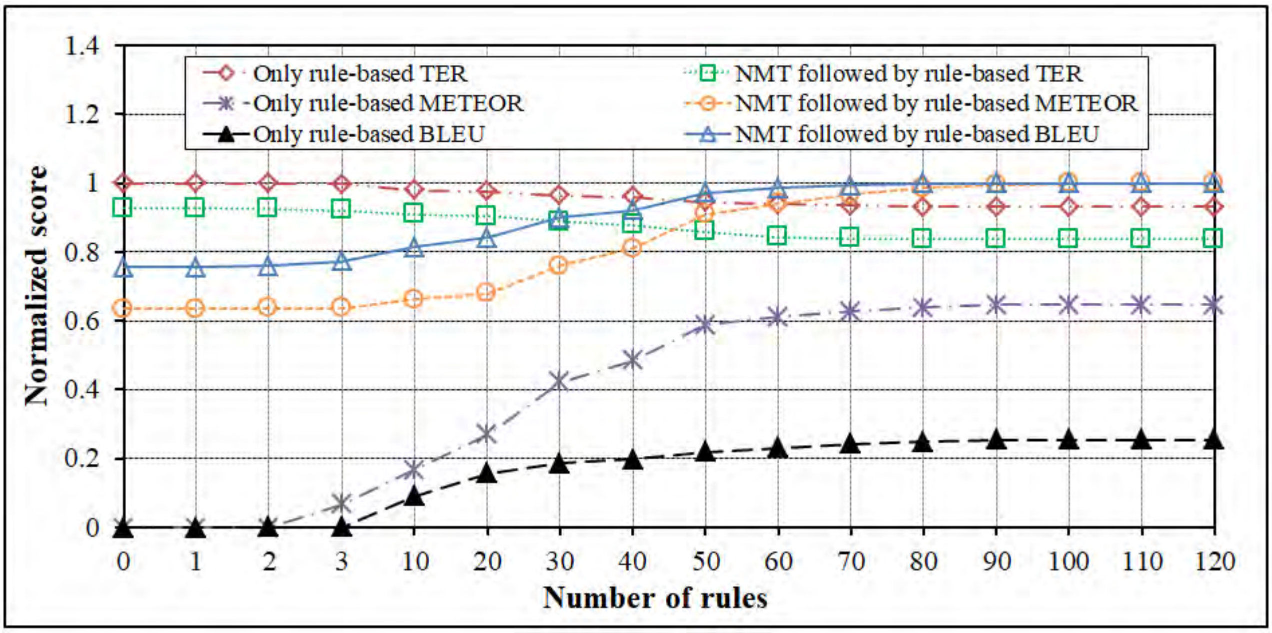


Figure 4.37: Comparison of normalized performance scores with an increase in the number of implemented rules for full dataset

translation of a sentence.

4.5 Resource Overhead

In this section, we analyze resource overheads required for our different translation approaches. Specifically, we cover time overhead and memory overhead in the subsequent sections.

4.5.1 Time Overhead

Time overhead increases as number of implemented rules increases in our system as shown in Figure 4.38. In this figure, we show time overhead per sentence translation for three different approaches with respect to an increase in the number of implemented rules - only rule-based approach, NMT, and ‘NMT followed by rule-based’ approach. Similar to the graphs for performance scores (METEOR and BLEU), the curves of only rule-based approach and ‘NMT followed by rule-based’ approach also exhibit significant rise for first 60-70 rules. However, unlike those (METEOR and BLEU) graphs, these two curves keep rising slowly rather than getting flat as we keep adding more rules. Besides, the curve for NMT remains flat (parallel to X axis) since time overhead of NMT does not change with

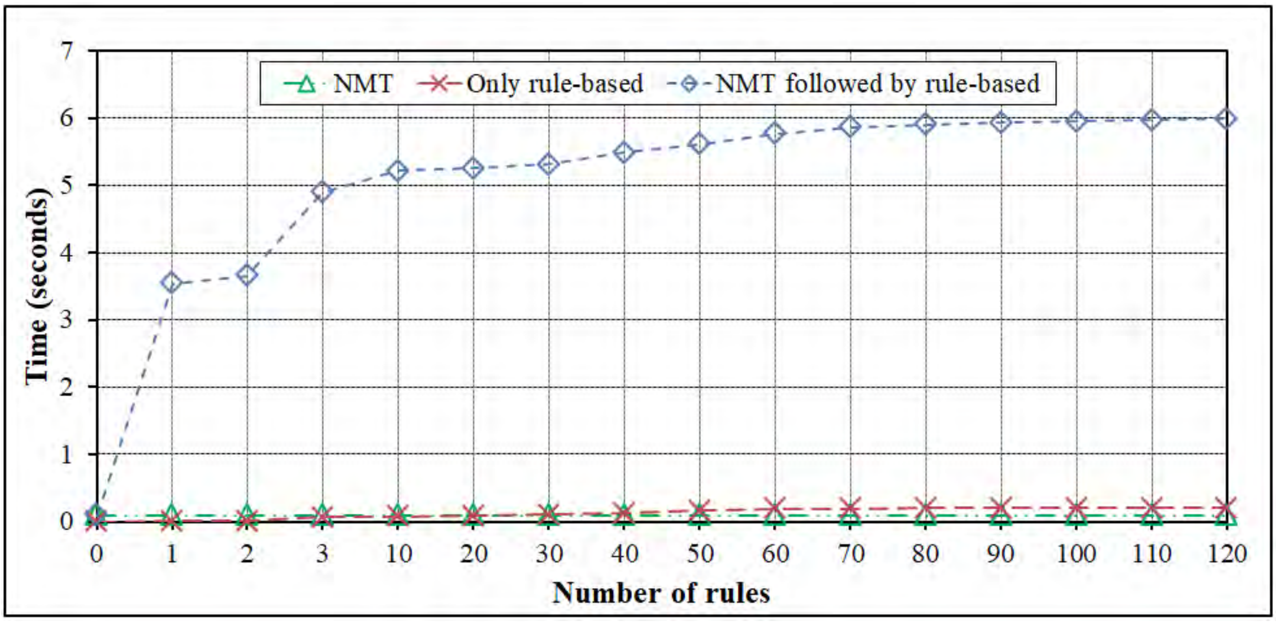


Figure 4.38: Comparison in variation of time with an increase in the number of implemented rules for literature-based dataset

the number of implemented rules.

Here, we calculate time overhead of 'NMT followed by rule-based' approach as the summation of time overheads of only rule-based approach, NMT, and blending between them. Table 4.10 shows a summary of time overheads of our aforementioned approaches for different number of implemented rules.

Number of rules	Rule-based	NMT	Blending	NMT followed by rule-based
3	67ms	90ms	4.743s	4.900s
10	77ms	90ms	5.053s	5.220s
50	163ms	90ms	5.357s	5.610s
70	196ms	90ms	5.574s	5.860s
120	204ms	90ms	5.699s	5.993s

Table 4.10: Time overheads of rule-based, NMT, and 'NMT followed by rule-based' for literature-based dataset

Next, we also explore time overheads of different approaches with respect to an increase in the number of implemented rules using our full dataset. Figure 4.39 shows three different curves (rule-based, NMT, and NMT followed by rule-based) generated using our full dataset. We notice that behaviour of each of the curves for this dataset remains identical to that of the curves for our previous

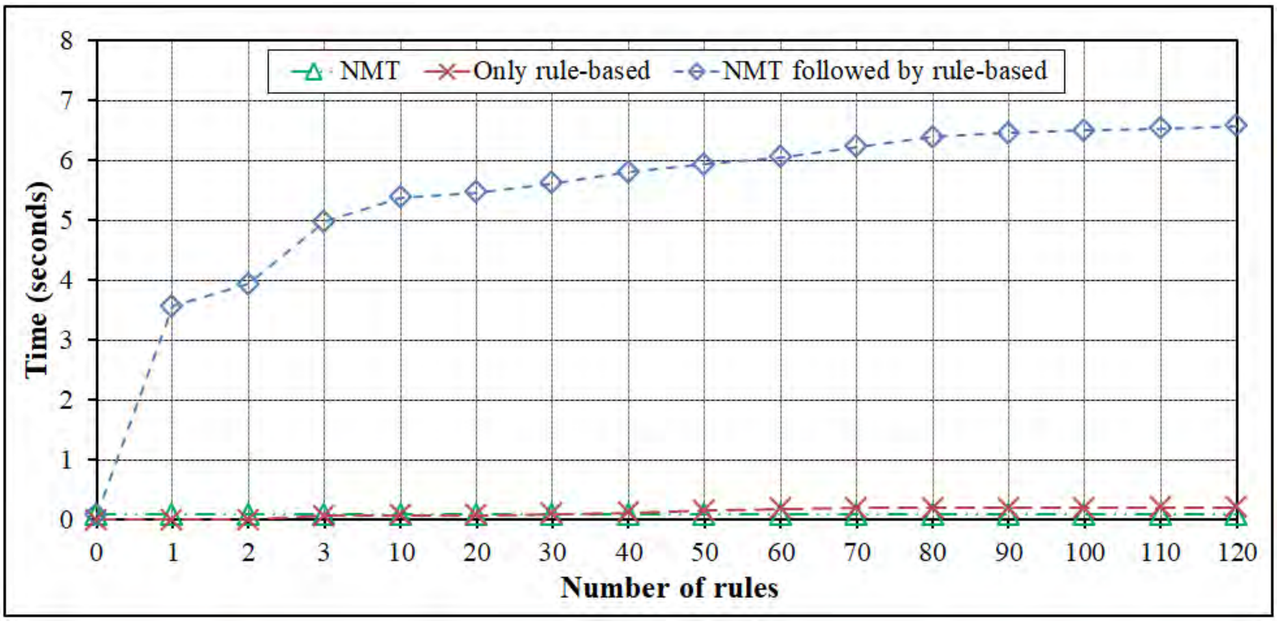


Figure 4.39: Comparison in variation of time with an increase in the number of implemented rules for full dataset

(literature-based) dataset.

4.5.2 Memory Overhead

In addition to that, we perform analysis on memory consumption overhead for our different approaches with respect to different number of rules. Behaviour of memory consumption curves is similar to that of time overhead curves as shown earlier. Figure 4.40 shows three curves (only rule-based approach, NMT, and 'NMT followed by rule-based' approach) reflecting total memory consumption per sentence translation with an increase in the number of implemented rules using our literature-based dataset. Here, the curves of only rule-based approach and 'NMT followed by rule-base' approach exhibit significant rise for first 60-70 rules, whereas the NMT curve remains flat (parallel to X axis). We consider unit of memory consumption as kilobytes (KB).

Next, we also explore memory consumption of different approaches with respect to an increase in the number of implemented rules using our full dataset. Figure 4.41 shows three different curves (rule-based, NMT, and NMT followed by rule-based) generated using this dataset.

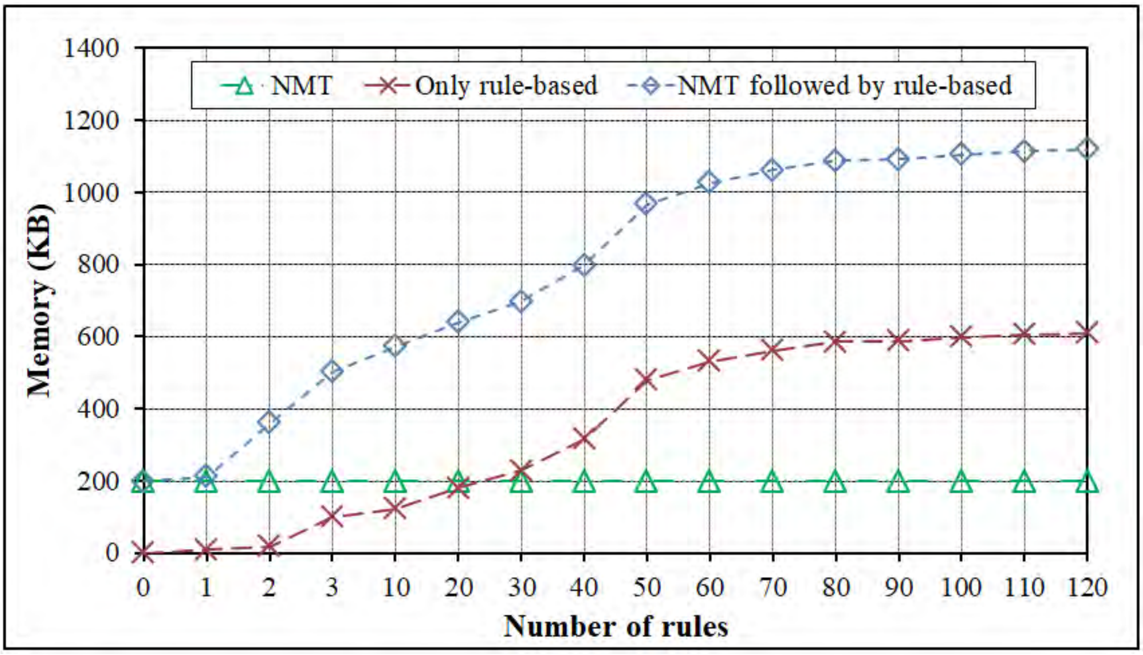


Figure 4.40: Comparison in variation of memory consumption with an increase in the number of implemented rules for literature-based dataset

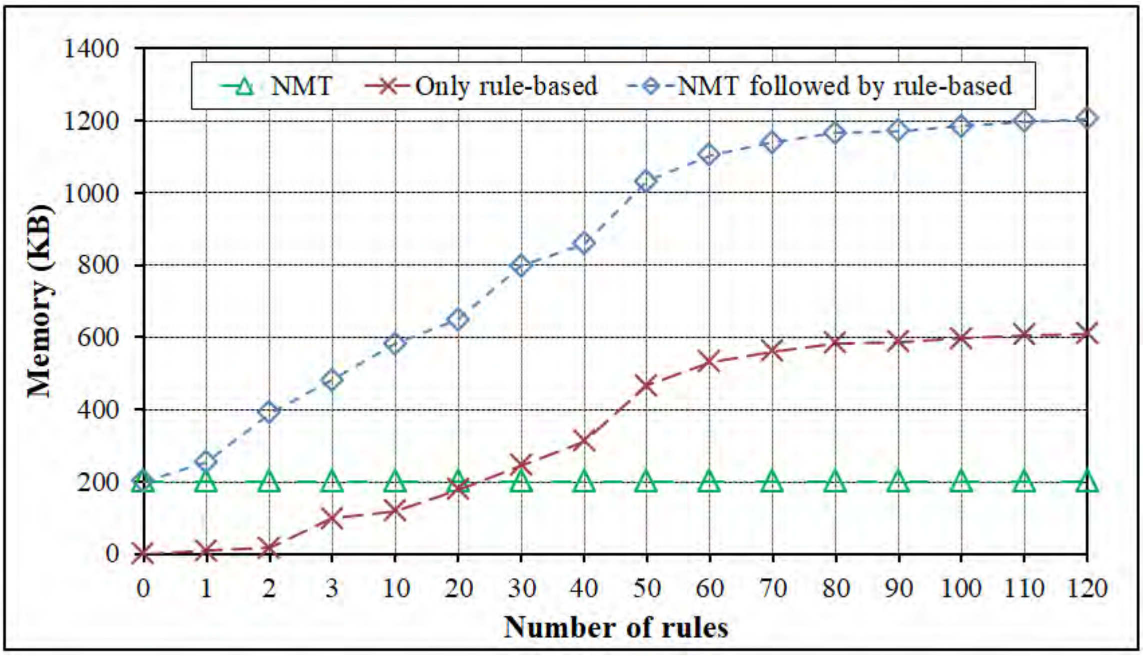


Figure 4.41: Comparison in variation of memory consumption with an increase in the number of implemented rules for full dataset

4.6 Overall Comparison

We summarize different results (performance scores, time overhead, and memory overhead) obtained using our different datasets in Table 4.11 and Table 4.12. Here, Table 4.11 reflects the results obtained for our literature-based dataset, and Table 4.12 reflects the results obtained for our full dataset.

	NMT	**Rule-based**	**NMT+rule-based**	**Rule-based+NMT**	**NMT or rule-based**
BLEU	8.56	1.28	11.43	0.84	8.80
METEOR	12.34	13.50	20.31	10.62	12.43
TER	93.73	93.90	85.09	96.62	93.50
Time (s)	0.090	0.204	5.993	4.011	0.806
Memory (KB)	200	610.982	1120.002	998.614	902.100

Table 4.11: Comparison among different translation approaches for literature-based dataset

	NMT	**Rule-based**	**NMT+rule-based**	**Rule-based+NMT**	**NMT or rule-based**
BLEU	9.28	3.13	12.26	1.34	9.87
METEOR	14.18	14.43	22.32	12.86	14.92
TER	92.78	93.21	83.83	95.52	92
Time (s)	0.092	0.203	6.569	4.97	0.807
Memory (KB)	200.60	609.702	1204.228	1037	903.341

Table 4.12: Comparison among different translation approaches for full dataset

4.7 Overall Experimental Findings

Next, we present our overall experimental findings in terms of average percentage (%) improvement of our different blending approaches over different parameters such as BLEU, METEOR, and TER in Table 4.13, Table 4.14, Table 4.15, and Table 4.16. Here, Table 4.13 and Table 4.14 reflect the results (average percentage (%) improvement) for literature-based dataset and full dataset respectively with

Parameters	**NMT+rule-based**	**Rule-based+NMT**	**NMT or rule-based**
BLEU	34%	-90%	3%
METEOR	65%	-14%	1%
TER	9%	-3%	0%

Table 4.13: Overall percentage (%) improvement over different parameters with respect to NMT for literature-based dataset

respect to NMT,. Note that we find these percentage improvements of our different approaches with respect to NMT.

Parameters	NMT+rule-based	Rule-based+NMT	NMT or rule-based
BLEU	32%	-86%	6%
METEOR	57%	-9%	5%
TER	10%	-3%	1%

Table 4.14: Overall percentage (%) improvement over different parameters with respect to NMT for full dataset

Similarly, Table 4.15 and Table 4.16 reflect the results (average percentage (%) improvement) for literature-based dataset and full dataset respectively with respect to only rule-based approach. Here, we find that % improvements of our different blending approaches with respect to rule-based approach is much higher than NMT approach.

Parameters	NMT+rule-based	Rule-based+NMT	NMT or rule-based
BLEU	793%	-34%	588%
METEOR	50%	-21%	-8%
TER	9%	-3%	0%

Table 4.15: Overall percentage (%) improvement over different parameters with respect to rule-based approach for literature-based dataset

Parameters	NMT+rule-based	Rule-based+NMT	NMT or rule-based
BLEU	292%	-57%	215%
METEOR	55%	-11%	3%
TER	10%	-2%	1%

Table 4.16: Overall percentage (%) improvement over different parameters with respect to rule-based approach for full dataset

4.8 Extension of Our Experimental Results

Machine translation is in practice for long time in different forms such as Example-based Machine Translation [4], Phrase-based Machine Translation [5], Statistical Machine Translation (SMT) [44], Neural Machine Translation (NMT) [50], etc. NMT is the most recent technology in machine translation, which outperforms all other translation approaches. This is why, we attempt to contribute in machine translation keeping NMT as our prime focus, and adopt NMT in our system. However, NMT depends largely on size and quality of dataset, which we lack significantly for Bengali language. Therefore, we extend our experimentation on another popular machine translation technology, Statistical Machine Translation (SMT). SMT was used by popular Google Translator just before NMT, not more than five years earlier.

Besides, M. Mumin et al., recently reported a Phrase-Based Statistical Machine Translation system between English and Bengali languages in both directions claiming to have achieved a promising BLEU score 17.43 for Bengali to English translation. In this regard, we adopt their baseline SMT system [37] to investigate the performance of SMT using our dataset. To do so, first, we implement a popular SMT toolkit, Moses [59], and we configure the system following their configuration process [37]. Next, we train the SMT system with our combined (literature-based and custom) dataset. Finally, we evaluate the performance of SMT using our dataset.

We achieve BLEU score 12.31 using the baseline SMT. In addition to that, we investigate the performance of our different blending approaches. We blend our rule-based translator with SMT this time. We present the performance scores of different approaches in Table 4.17.

Score	**SMT**	**Rule-based**	**SMT+rule-based**	**Rule-based+SMT**	**SMT or rule-based**
BLEU	12.31	3.13	16.43	2.16	14.14
METEOR	15.35	14.43	22.33	13.48	20.92
TER	88.14	90.17	82	93.35	85.38

Table 4.17: Comparison among different translation approaches considering SMT as baseline system

Table 4.17 reflects that our ‘SMT or NMT followed by rule-based’ approach stills remains the best translation approach. Interestingly, performance of ‘SMT followed by rule-based’ approach (BLEU = 16.43) is better than ‘NMT followed by rule-based’ approach (BLEU = 12.26) since in this case, SMT (BLEU = 12.31) performs better than NMT (BLEU = 9.28) in isolation. This happens because, our dataset is not large enough to train an NMT system efficiently. SMT perhaps takes this advantage to outperform NMT by a small margin for this dataset. Besides, our best approach (BLEU = 16.43) lags behind their proposed approach (BLEU = 17.43) in terms of overall performance score because of our insufficient dataset again. They trained their system with a large dataset containing 197,338 parallel Bengali-English sentences, which is more than 16 times larger than our current dataset. However, their dataset is not made publicly available.

Nonetheless, SMT scores 16.91 using their dataset [37], whereas SMT scores 12.26 using our dataset. They achieve BLEU score 17.43 in their approach over SMT score 16.91 [37], which offers an improvement of 3.1% over SMT. However, our best translation approach achieves BLEU score 16.43 over SMT score 12.31, which offers 33.5% improvement over SMT. Therefore, we expect to achieve much higher BLEU score when we can match their dataset in future. In addition to that, this extended experimentation leads to an important finding - "Any translation generated by machine (NMT or SMT) can be significantly improved after blending with rule-based translator".

4.9 Extending Our Study to A High-Resource Context

Performance of data-driven translators (NMT or SMT) largely depends on availability of significant amount of training data. However, the largest dataset used in our experimentation presented so far consists of up to 11,500 parallel Bengali-English sentences. Only 11,500 sentences may not really satisfy the need for significant amount of training data for an NMT system mimicking the context of low-resource language.

However, we are yet to show what would happen if we take our approach to a high-resource context. Therefore, we extend our study to a high-resource context by developing a larger Bengali-English parallel corpus containing more than one million sentence pairs. We summarize the performance scores of our different approaches obtained using this dataset in Table 4.18. Table 4.18 again establishes that our 'NMT followed by rule-based' approach performs the best over all other alternative approaches.

Score	**NMT**	**Rule-based**	**NMT+rule-based**	**Rule-based+NMT**	**NMT or rule-based**
BLEU	13.43	4.16	18.73	2.89	14.51
METEOR	24.82	16.22	31.30	13.65	26.87
TER	85.79	88.50	77.94	92.84	83.14

Table 4.18: Comparison among different translation approaches for a high-resource context

Next, we present the improvement in performance scores of all the approaches with respect to an increase in the size of dataset in Figure 4.42. The figure shows that performance improves with an increase in the size of dataset. Here, we also show a comparison between NMT and 'NMT followed by rule-based' approach in terms of BLEU scores using our different datasets. Note that we find the best performance score (BLEU = 18.73) after extending our experimentation to the high-resource context (with one million sentence pairs), which is substantially higher than our previous best score (BLEU = 12.43) obtained for the low-resource context (with 11,500 sentence pairs).

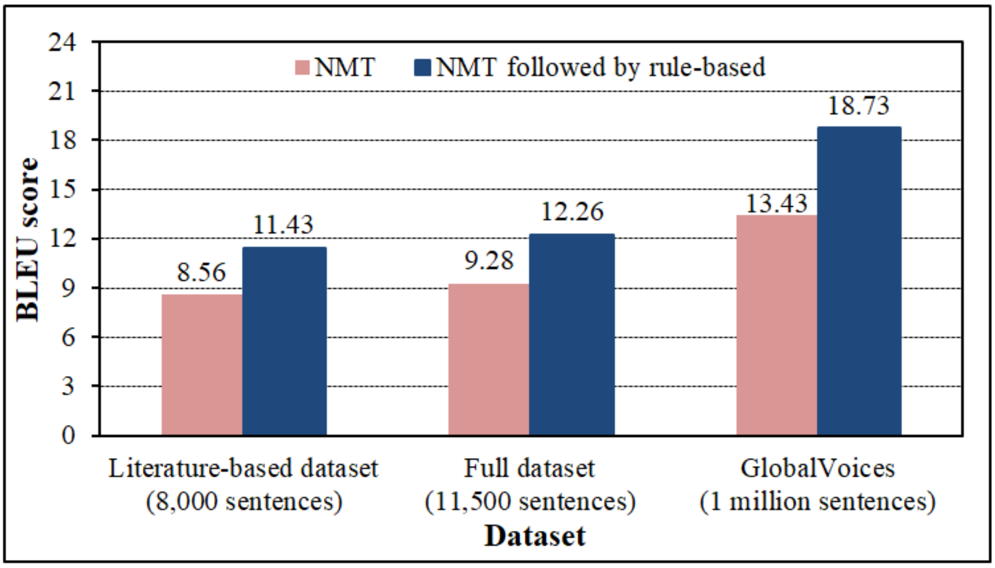


Figure 4.42: Comparison between NMT and 'NMT followed by rule-based' approach in terms of BLEU scores with different datasets

Finally, we present the improvement in performance scores of all the approaches with respect to an increase in the number of training steps in Figure 4.43. The figure shows that performance improves as we increase the number of steps for training the data-driven translators.

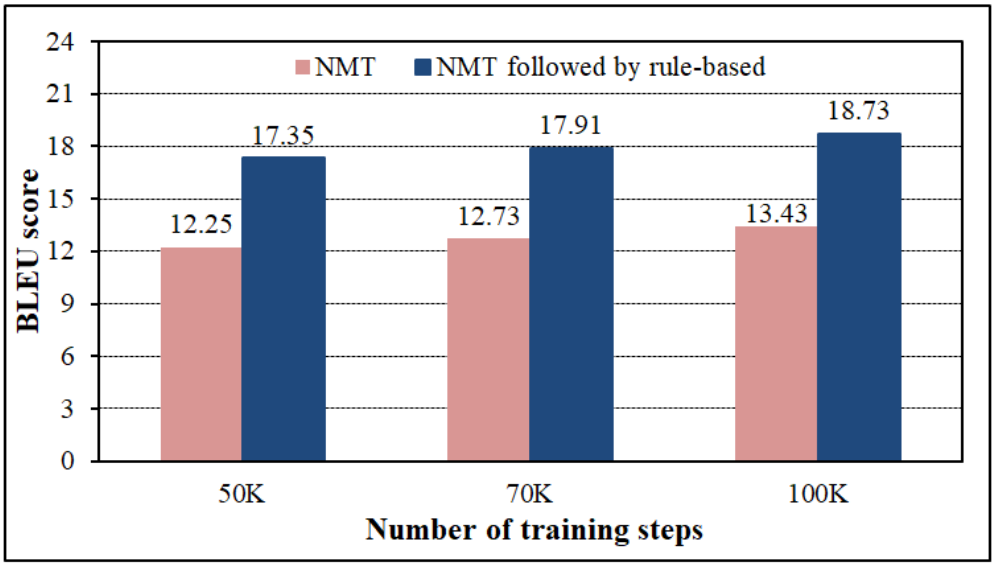


Figure 4.43: Comparison between NMT and 'NMT followed by rule-based' approach in terms of BLEU scores with respect to an increase in the number of training steps

Chapter 5

Analogy to Human Behaviour: A Casual Cross Checking to Our Proposed Methods and Their Results

At this point, we perform a casual cross checking to our proposed methods and their results with respect to human behaviour. Note that the idea of our proposed translation approaches actually comes from how people approach translation in real life. In this regard, we conduct a survey to identify how people generally perform translation from one language to another such as from Bengali to English. Around 150 participants respond to this survey by sharing their own translation approach. The survey basically presents a very simple question to the participants on how they perform translation from one language to another (Bengali to English in our case). As the possible answers to the question, we provide all possible options. Thus, we form the question as follows.

Question: How do you prefer to translate from one language to another language (for example, Bengali to English)?

□ Use experience only (how others speak and reading bilingual books) with no formal grammatical knowledge

□ Strictly stick to applying knowledge on grammatical rules only

□ Apply both grammatical rules and experience from various sources in any order

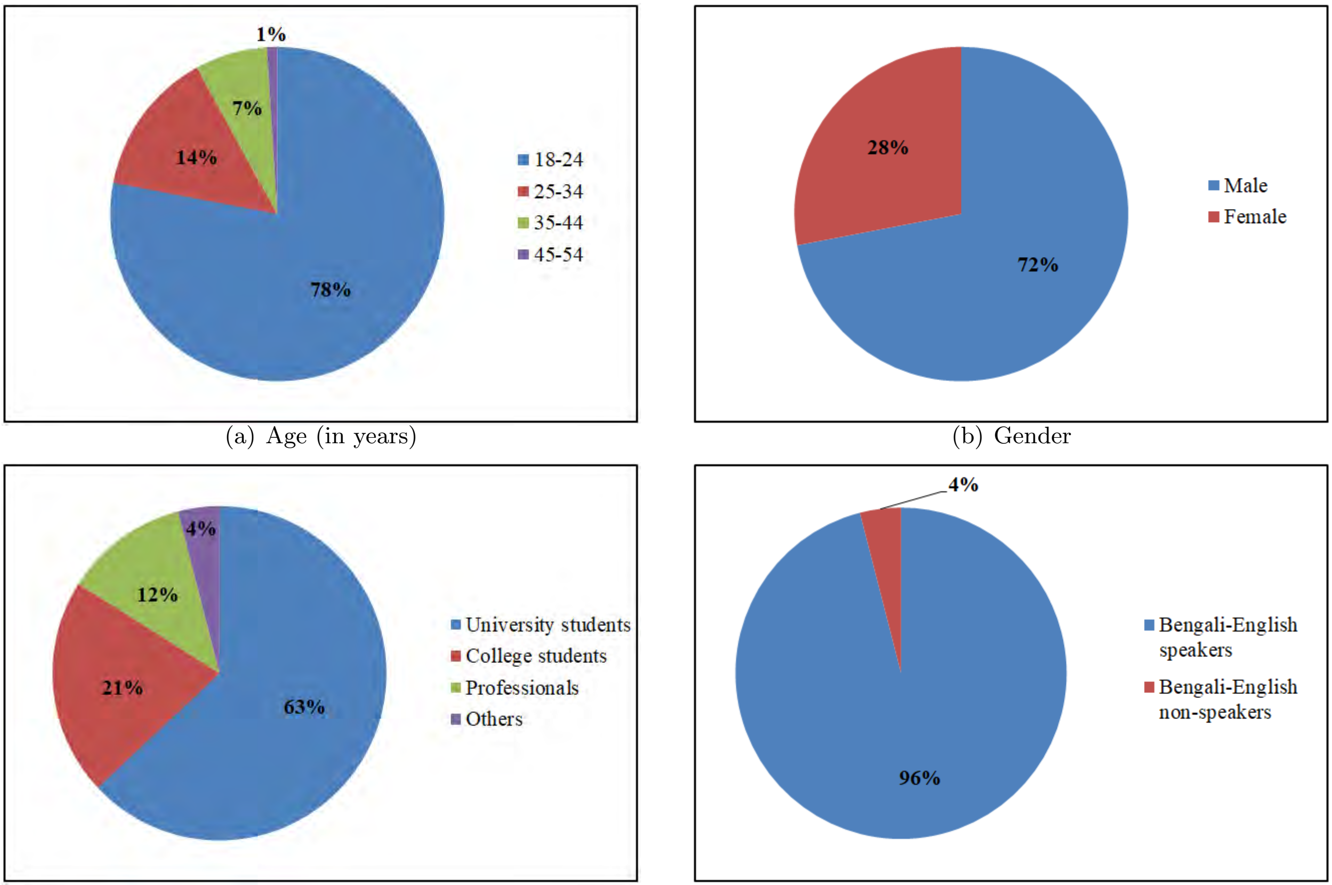
□ Apply formal grammatical rules first to translate initially, and then try to use experience (on you have seen or heard something like your initially translated sentence) to modify it (may be slightly) to get more accurate translation

□ Apply your experience first, then apply grammatical checking to make the translated sentence more accurate

□ Translate separately using only grammatical rules and only experience. Then, decide any one of them without mixing one with another at all

5.1 Demography of Survey Participants

Next, we present the demography of the participants. People from different ages, genders, and backgrounds take part in our survey. Besides, both Bengali-English speakers and non-speakers respond to our survey. We present the demography of participants in Figure 5.1.



(a) Age (in years)

(b) Gender

(c) Background

(d) Speaking language

Figure 5.1: Demography of survey participants

The figure shows that our participants cover different ages (Figure 5.1(a)), genders (Figure 5.1(b)), backgrounds (Figure 5.1(c), and speaking languages (Figure 5.1(d). We note that majority of the participants are male students, aged in between 18-24 years. In addition to that, people having no formal educational background (4%) also take part in our survey.

5.2 Survey Results

Majority of the survey participants respond that they use experience first before applying grammatical rules to translate from one language to another. This process mimics our 'NMT (or SMT) followed by rule-based' translation approach. This happens as experience generally refers to learning how others communicate or speak in the target language along with reading materials in that language. It is similar to the learning process of our mother tongue. Here, we interpret 'using experience' as 'translation generated by machine (NMT or SMT)' since machine translates based on its learning acquired through rigorous training with corpus (datasets). Through using the experience, a forma of the target sentence generally gets generated that is also done by the NMT (or SMT). Therefore, the case of using experience first, then grammatical rules is analogous to our 'NMT (or SMT) followed by rule-based' case. Besides, some of our survey participants also prefer to translate by applying rules first, then experience, which is analogous to our 'rule-based followed by NMT' approach. We show a mapping between different types of human translation approaches (considered in our survey) and our proposed translation approaches in Table 5.1. Next, we present the results obtained from our survey in Figure 5.2. Therefore, we find that this survey result supports our obtained experimental results since our results also imply that 'NMT (or SMT) followed by rule-based' approach is the best translation approach in machine translation.

Human translation approach	**Our proposed translation approach**
Only experience	NMT (or SMT)
Only rules	Rule-based
Experience and rules in any order	Not applicable
Rules first, then experience	Rule-based followed by NMT (or SMT)
Experience first, then rules	NMT (or SMT) followed by rule-based
Either experience or rules	Either NMT (or SMT) or rule-based
Others (specify)	Not applicable

Table 5.1: Mapping between human translation approaches and our proposed translation approaches

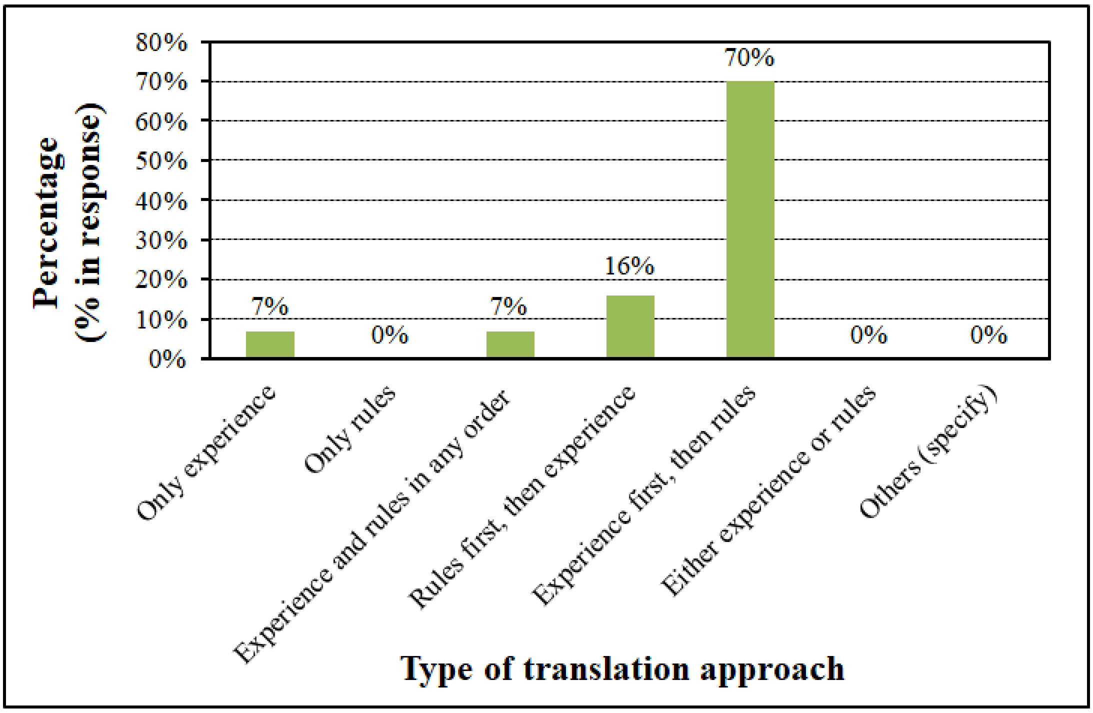


Figure 5.2: Results of survey participants' responses

Chapter 6

Avenues for Further Improvements

People from various backgrounds thrive for learning multiple languages equal effectively for sustaining in the era of technology and communication. Translators offer great help to accomplish such a laborious task. However, very basic and primary sentence building rules of any language usually consist of a good number of exceptions. Keeping track of wide varieties of such possible cases is one of the most challenging tasks of a translation system, even the most intelligent living beings are not any exception to it. For example, let us consider these two sentences:

1 “The complex houses married and single soldiers and their families.” This is what is called a garden path sentence. Though grammatically correct, the readers initial interpretation of the sentence may be nonsensical. Here, ‘complex’ may be interpreted as an adjective and ‘houses’ may be interpreted as a noun. Readers are immediately confused upon reading that ‘the complex houses married’, interpreting ‘married’ as the verb. How can houses get married? In actuality, ‘complex’ is the noun, ‘houses’ is the verb, and ‘married’ is the adjective. The sentence is trying to express the following: “Single soldiers, as well as married soldiers and their families, reside in the complex.”

2 “All the faith he had had had had no effect on the outcome of his life.” This sentence is an example of lexical ambiguity. As strange as this sentence might sound, it is actually grammatically correct. The sentence relies on a double use of the past perfect. The two instances of ‘had had’ play different grammatical roles in the sentences. The first one is a modifier while the second one is the main verb of the sentence.

Because of the presence of such ambiguities in all languages, even the most sophisticated software cannot substitute the skill of a professional translator. Besides, the reasons why machine translations are not as satisfactory as human translation are many. One of the reasons that translators cannot replace professional human translators is the same reason that plain old bilingual laypeople cannot replace professional human translators for many tasks. For most of the translation jobs, the task requires more than just knowledge of two languages. Translators can not be walking dictionaries. They need to recreate language by crafting beautiful phrases and sentences to make them have the same impact as the source. Often, they may need to devise brand new ways of saying things by translation, and to do so, they must draw upon a lifetime's worth of knowledge derived from living in two cultures. However, machine translators cannot exactly do that. Considering all these apparently unavoidable limitations of a machine translator, we must also accept that machine translation is now vital to the top industries around the world, and one of the most promising fields in research sector. However, this topic is very little explored for low-resource languages such as Bengali, which opens up scope for large varieties of possible future work in this research area.

6.1 Future Work

Things are changing fast in the world of translation technology. As each year passes, improvements in computational capacity, AI, and data analysis expand in terms of both speed and accuracy of machine translation. Although previous forms of machine translation were completely rule-based (RBMT) or phrase-based (PBMT), NMT makes the translation process look less like a computer, and more like a human. However, the road to replace human translators with NMT may be a long one. We discuss some of our possible future work as follows:

- Although NMT is a huge success in translation industry, it does not perform equally well for all languages. Different languages can vary from each other to a great extent in terms of word embeddings, inferences, etc. We plan to create an efficient word embedding module for Bengali soon.

- Although we achieve improvement over classical NMT in terms of performance scores, it comes at a cost of higher resource (time and memory) overheads than NMT. Therefore, we plan to optimize the resource overheads required for our proposed approaches.

- We plan to explore other possible modes of blending such as phrase-based blending, trained blending, morphological blending, etc., in future.

- Efficient AI techniques, indexing and searching mechanisms will improve the total system that may result in more accurate output. We plan to devise a more efficient algorithm for token tagging and searching words in vocabulary.
- One of the main challenges in Bengali to English text conversion remains in implementing its vast grammatical set of rules. If we can track more core rules to overshadow the ambiguous grammars then the translation task will be simpler and compact. Therefore, we plan to standardize and optimize the set of implemented rules for Bengali in our rule-based translator.
- Building a parallel corpus for Bengali-English sentence pairs is one of the most demanding tasks for translation of Bengali sentences using NMT. While other high-resource languages have available parallel corpus containing millions of sentence pairs, there is no such corpus for Bengali even containing thousand sentences which drastically degrades the translation performance of Bengali sentences using NMT.
- There are lots of research opportunities in language processing sector. Since languages keep evolving continuously, we need to find a way to update new grammatical rules. Machine Learning using Statistical MT can be one way. We plan to investigate integration of statistical language model with our rule-based model for future improvement.
- Role of prepositions in a sentence can be ambiguous. Therefore, another idea of our future work is to extend the preposition handling component. Besides, adding more postpositional words and inflectional suffixes would improve the system's translation performance.
- Developing Opennlp tools for parts-of-speech tagging of Bengali words in a sentence efficiently is one of the most crucial and less explored tasks in Bengali to English translation. Currently, there is an efficient Opennlp tool for parts-of-speech tagging of words in English sentences. We aim to extend our work on developing Opennlp tools for Bengali language which will definitely create a landmark in Bengali language processing.
- Finding applications of WordNet in different areas of NLP. We plan to develop a WordNet for Bengali in future.

Chapter 7

Conclusion

Millions of immigrants thrive for working knowledge on popular non-native languages such as English, as this creates many opportunities in international communities. Translators can offer a great help to accomplish such a laborious task. On the other hand, in case of machine translation, NMT has emerged as the most promising approach in recent years. NMT mostly outperforms all other previous translation technologies. Google Translator, one of the most popular and widely available translators, also uses NMT approach for translating from one language to another. However, NMT-based systems perform poorly for translating low-resource languages such as Bengali, Arabic, etc. Therefore, the importance of an efficient translator for such languages is noteworthy.

Bengali, being one of the most popular and widely-spoken languages worldwide, remains little explored in some crucial areas of machine translation research. Existing research studies in this regard mostly focus on English to Bengali translation, as only a handful studies have been performed on translating from Bengali to English. Besides, although some of the existing studies focus on rule-based translation for translating from Bengali to English, these studies lack in processing Bengali words semantically from various aspects such as finding stems of different forms of Bengali verbs, processing unknown words, etc. Moreover, to the best of our knowledge, none of the studies existing in the literature focuses on integration between rule-based translator and data-driven machine translators such as NMT, SMT, etc. Accordingly, we focus on all these yet to be focused aspects in our study.

In our study, we make our contribution from three perspectives. First, we develop and implement a new rule-based translator from the scratch, which covers several basic grammatical rules for Bengali to English translation. Our rule-based translator adopts new methodologies for stemming of Bengali verbs and processing unknown words. Second, we separately incorporate two popular data-driven machine translation approaches (NMT and SMT). Finally, we explore different possible approaches for blending these two translation schemes (rule-based translation and data-driven machine translation). We also evaluate performance of each of the blending approaches in terms of standard translation performance metrics.

As revealed in our study, a number of critical issues always make natural language processing and translation tasks more complex. For a rule-based translator, there remain a number of exceptions that violate the standard rules of grammar, which are quite tough to tackle by implementing any number of rules [57]. Hence, the efficiency of a rule-based translator in translating languages with complex grammatical structures is very low. On the other side of the coin, translations generated by data-driven machine translator can be unreliable, offensively wrong, or utterly unintelligible sometimes [10]. Besides, such machine translation systems have a steeper learning curve with respect to the amount of training data, resulting in worse quality in low-resource settings. Thus, the performance of a rule-based translator is constrained by the number of incorporated rules whereas the performance of a data-driven translator is constrained by the amount of data fed to it for learning or training. In reality, it is very difficult to ensure sufficiency either in terms of the number of rules or in terms of the amount of data. Accordingly, neither of the two different types of approaches can suffice all alone.

Considering these realistic aspects, we explore different approaches of blending between rule-based translator and data-driven machine translator (NMT and SMT) to investigate whether and how a synergy between these translators can be attained. Here, we mainly focus how the different types of translators can work in combination rather than in isolation. Our study leads to some promising outcomes as two of our blending approaches outperform both NMT and SMT in isolation going much beyond the rule-based translator. In addition to exploring the blending approaches, we also investigate how our rule-based translator (for translating from Bengali to English) can be made more efficient in isolation.

While conducting our study, we have found that it is extremely difficult (if not impossible) to get a large parallel corpus for Bengali to English translation. Accordingly, we plan to work on building such a corpus in future. Besides, we will also focus on improving the neural network level architecture used in the NMT considering specific aspects of translating from Bengali to English. Besides, we also plan to limit resource usage required for our blending purpose. In addition to that, we plan to explore other possible modes of blending such as phrase-based blending, trained blending, etc., in future. Finally, exploring our proposed blending approaches for other language pairs remains yet another future work of this study.

Bibliography

[1] O. Bozar, C. Federmann, M. Fishell, Y. Graham, B. Haddow, and M. Huck, "Findings of the 2018 Conference on Machine Translation (WMT18)", in Proceedings of the Third Conference on Machine Translation (WMT), vol. 2, pp. 272-303, ACL, 2018.

[2] S. Bal, S. Mohanta, L. Mondal, and R. Parekh, "Bilingual Machine Translation: English to Bengali", in Proceedings of the International Ethical Hacking Conference, pp. 247-259, Springer, 2018.

[3] M. Rahman, M. F. Kabir, and M. N. Huda, "A Corpus Based N-gram Hybrid Approach of Bengali to English Machine Translation", in Proceedings of the International Conference on Computer and Information Technology (ICCIT), pp. 1-6, IEEE, 2018.

[4] M. Roy, "A Semi-supervised Approach to Bengali-English Phrase-Based Statistical Machine Translation", in Proceedings of the Canadian Conference on Artificial Intelligence, pp. 291-294, Springer, 2009.

[5] R. Gangadharaiah, R. D. Brown, and J. G. Carbonell., "Phrasal equivalence classes for generalized corpus based machine translation", in Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, pp. 13 - 28, Springer, 2011.

[6] J. D. Kim, R. D. Brown, and J. G. Carbonell, "Chunk-Based EBMT", in Proceedings of the 14th Annual Conference of European Association for Machine Translation, pp. 1-8, MT Archive, 2010.

[7] 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), pp. 648-653, IEEE, 2004.

[8] M. K. Rahman and N. Tarannum, “A Rule Based Approach for Implementation of Bangla to English Translation”, in Proceedings of the International Conference on Advanced Computer Science Applications and Technologies (ACSAT), pp. 13-18, IEEE, 2012.

[9] Y. Wu, M. Schuster, Z. Chen, and Q. V. Le, “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”, in Proceedings of the Computing Research Repository, pp.1-23, arXiv, 2016.

[10] P. Koehn and R. Knowles, “Six Challenges for Neural Machine Translation”, in Proceedings of the Computing Research Repository, pp. 1-12, arXiv, 2017.

[11] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, “Unsupervised Neural Machine Translation”, in Proceedings of the International Conference on Learning Representations (ICLR), pp. 1-12, OpenReview, 2018.

[12] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, “Unsupervised machine translation using monolingual corpora only”, in Proceedings of the International Conference on Learning Representations (ICLR), pp. 1-14, OpenReview, 2018.

[13] A. Haque, A. Islam, and A. B. M. A. A. Islam, “An Approach Towards Multilingual Translation By Semantic-Based Verb Identification And Root Word Analysis”, in Proceedings of the 5th International Conference on Networking, Systems and Security (NSysS), pp. 1-9, IEEE, 2018.

[14] A. Islam, A. Haque, and A. B. M. A. A. Islam, “Polyglot: An approach towards reliable translation by name identification and memory optimization using semantic analysis”, in Proceedings of the 4th International Conference on Networking, Systems and Security (NSysS), pp. 1-8, IEEE, 2017.

[15] M. Islam and A. B. M. A .A. Islam, “Polygot: Going Beyond Database Driven And Syntax-based Translation”, in Proceedings of the 7th Annual Symposium on Computing for Development, pp. 28-31, ACM, 2016.

[16] A. Klementiev, A. Irvine, C. CallisonBurch, and D. Yarowsky, “Towards statistical machine translation without paralel corpora”, in Proceedings of the 13th Conference of European Chapter of the Association for Computational Linguistics (EACL), pp. 130-140, ACL, 2012.

[17] E. Ristad and P. Yianilos, “Learning String Edit Distance”, in Proceedings of the International Conference on Pattern Analysis and Machine Intelligence, vol. 20, pp. 522 - 532, IEEE, 1998.

[18] R. Haldar and D. Mukhopadhyay, “Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach”, in Proceedings of the Computing Research Repository, pp. 1-5, arXiv, 2011.

[19] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation”, in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311-318, ACL, 2002.

[20] S. Banerjee and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgements”, in Proceedings of the ACL workshop, pp. 65-72, ACL, 2005.

[21] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhou, “A Study of Translation Edit Rate with Targeted Human Annotation”, in Proceedings of the Association for Machine Translation in the Americas, pp. 223-231, ACL, 2006.

[22] S. Dandapat, S. Sarkar, and A. Basu, “Automatic parts-of-speech tagging for Bengali: an approach for morphologically rich languages in a poor resource scenario”, in Proceeding of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 221-224, ACL, 2007.

[23] S. A. Chowdhury, “Developing a Bangla to English Machine Translation System Using Parts Of Speech Tagging: A Review, Journal of Modern Science and Technology, vol. 1. no. 1., pp. 113-119, JMST, 2013.

[24] M. H. Haque, M. F. Hossain, and A. F. Hossain, “Machine and Web Translator for English to Bangla using Natural Language Processing”, Daffodil International University Journal Of Science & Technology, vol. 5, no. 1, pp. 53-61, DIUJST, 2010.

[25] H. Khoshnoudi, “Investigating the Quality of the Translations of Quran through Equivalence Theory: A Religious Lexicology of the Word Roshd”, International Journal of English Language & Translation Studies, vol. 7, no. 3, pp. 19-24, ELTS Journal, 2019.

[26] S. K. Borhan, M. Hossain, and K. Biswas, “Bangla to English Text Conversion using opennlp Tools”, Daffodil International University Journal Of Science & Technology, vol. 8, no. 1, pp. 37-42, DIUJST, 2013.

[27] G. Foster, C. Goutte, and R. Kuhn, “Discriminative instance weighting for domain adaptation in statistical machine translation”, in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 451-459, ACM, 2010.

[28] D. Saha, S. K. Naskar, S. Bandyopadhyay, “A Semantics-based English-Bengali EBMT System for translating News Headlines”, in Proceedings of MT Summit-X, pp. 125-133, Asia-Pacific Association for Machine Translation, 2005.

[29] M. D. Huq, “Semantic values in Translating from English to Bangla”, Dhaka University Journal of Linguistics, vol. 1, no. 2, pp. 45-66, DUJL, 2009.

[30] G. Doddington, “Automatic Evaluation of Machine Translation Quality Using N-gram CoOccurrence Statistics”, in Proceedings of the 2nd International Conference on Human Language Technology Research, pp. 138-145, ACM, 2002.

[31] S. K. Naskar and S. Bandyopadhyay, “A Phrasal EBMT System for Translating English to Bengali”, in Proceedings of the International Conference on Language, Artificial Intelligence, and Computer Science for Natural Language Processing Applications (LAICS–NLP), pp. 372-379, ArXiv, 2005.

[32] M. M. Anwar, M. Z. Anwar, and M. A. Bhuiyan, “Syntax Analysis and Machine Translation of Bangla Sentences”, International Journal of Computer Science and Network Security, vol. 09, no. 08, pp. 317–326, IJCSNS, 2009.

[33] D. Saha, S. K. Naskar, and S. Bandyopadhyay, “A Semantics-based English-Bengali EBMT System for translating News Headlines”, in Proceedings of the 10th International MT Xummit, pp. 125-133, Asia-pacific Association for Machine Translation (AAMT), 2005.

[34] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning”, in Proceedings of the 25th International Conference on Machine learning (ICML ’08), pp. 160-167, ACM, 2008.

[35] M. Peters, M. Neumann, M. Lyyer, M. Gardner, and L. Zettlemoyer, "Deep Contextualized Word Representations", in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 2227-2237, ACL, 2018.

[36] K. Kann, K. Cho, and S. Bowman, "Towards Realistic Practices In Low-Resource Natural Language Processing: The Development Set", in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1-8, ACL, 2019.

[37] M. Mumin, M. Seddiqui, M. Iqbal, and M. Islam, "Shu-torjoma: An English<->Bangla Statistical Machine Translation System", Journal of Computer Science, vol. 15, no. 7, pp. 1022-1039, Science Publications, 2019.

[38] G. Haffari, M. Roy, and A. Sarkar, "Active learning for statistical phrase-based machine translation", in Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 415-423, ACM, 2009.

[39] D. Mimmo, H. M. Wallach, J. Naradowsky, D. Smith, and A. McCallum, "Poly-lingual topic models", in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '09), pp. 880-889, ACL, 2009.

[40] E. Alfonseca, M. Ciaramita, and K. Hall, "Gazpacho and summer rash: lexical relationships from temporal patterns of web search queries", in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1046–1055, ACL and AFNLP, 2009.

[41] J. Carbonell, S. Klein, D. Miller, M. Steinbaum, T. Grassiany, and J. Frey, "Context-Based Machine Translation", in Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pp. 19-28, AMTA, 2006.

[42] F. Och and H. Ney, "The alignment template approach to statistical machine translation", Journal of Computational Linguistics, vol. 30, no. 4, pp. 417-449, MIT Press, 2004.

[43] F. Och and H. Ney, "A systematic comparison of various statistical alignment models", Journal of Computational Linguistics, vol. 29, no. 1, pp. 19-51, MIT Press, 2003.

[44] P. Koehn, F. Och, and D. Marcu, “Statistical phrase-based translation”, In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL ’03), pp. 48-54, ACL, 2003.

[45] Google Translate, https://translate.google.com, last accessed on July 15, 2019.

[46] World Population Clock: 7.7 Billion People (2019) - Worldometers, www.worldometers.info, last accessed on March 31, 2019.

[47] Ethnologue, https://www.ethnologue.com/guides/how-many-languages, last accessed on June 19, 2019.

[48] Importance of learning english essay, https://friedpapers.com/essay/importance-of-learning-english-essay, last accessed on June 21, 2019.

[49] OpenNLP, www.maxnet.sourceforge.net, last accessed on March 13, 2019.

[50] NMT with Tensorflow, https://github.com/tensorflow/nmt, last accessed on June 15, 2019.

[51] Zipf’s Law and Heap’s Law, www.ccs.neu.edu., last accessed on June 30, 2019.

[52] Kaggle, https://www.kaggle.com/zusmani/the-holy-quran, last accessed on June 30, 2019.

[53] Prothom Alo, https://www.prothomalo.com, last accessed on June 30, 2019.

[54] Subtitles, https://www.subscene.com, last accessed on July 2, 2019.

[55] SUST website, https://www.sust.edu/d/cse/research, last accessed on July 10, 2019.

[56] History and rule-based system, https://www.inf.ed.ac.uk/teaching/courses/mt/lectures/history.pdf, last accessed on September 9, 2019.

[57] Ambiguous Grammar, https://www.thoughtco.com/syntactic-ambiguity-grammar-1692179, last accessed on September 9, 2019.

[58] English Idioms, https://www.ef.com/wwen/english-resources/english-idioms, last accessed on September 9, 2019.

[59] Moses, http://www.statmt.org/moses, last accessed on October 15, 2019.

[60] 3 reasons why neural machine translation is a breakthrough, https://slator.com/technology/3-reasons-why-neural-machine-translation-is-a-breakthrough, last accessed on September 10, 2019.

[61] GlobalVoices, http://opus.nlpl.eu/GlobalVoices.php, last accessed on September 10, 2019.