Shahjalal University of Science and Technology

Department of Computer Science and Engineering

Linguistic Analysis of English-Bangla Machine Translation

Masum Ahmed
Reg. No.: 2016331028
4th year, 2nd Semester

Md Shamihul Islam Khan
Reg. No.: 2016331078
4th year, 2nd Semester

Department of Computer Science and Engineering

Supervisor

Dr. Mohammad Abdullah Al Mumin
Professor
Department of Computer Science and Engineering

7th July, 2021

Linguistic Analysis of English-Bangla Machine Translation

A Thesis submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

By

Masum Ahmed
Reg. No.: 2016331028
4th year, 2nd Semester

Md Shamihul Islam Khan
Reg. No.: 2016331078
4th year, 2nd Semester

Department of Computer Science and Engineering

Supervisor

Dr. Mohammad Abdullah Al Mumin

Professor

Department of Computer Science and Engineering

7th July, 2021

Recommendation Letter from Thesis Supervisor

The thesis entitled *Linguistic Analysis of English-Bangla Machine Translation* submitted by the students

1. Masum Ahmed
2. Md Shamihul Islam Khan

is under my supervision. I, hereby, agree that the thesis/project can be submitted for examination.

Signature of the Supervisor:

Name of the Supervisor: Dr. Mohammad Abdullah Al Mumin

Date: 7th July, 2021

Certificate of Acceptance of the Thesis

The thesis entitled *Linguistic Analysis of English-Bangla Machine Translation* submitted by the students

1. Masum Ahmed
2. Md Shamihul Islam Khan

on 7th July, 2021 , hereby, accepted as the partial fulfillment of the requirements for the award of their Bachelor Degrees.

Head of the Dept.	Chairman, Exam. Committee	Supervisor
Dr. Mohammad Abdullah Al Mumin	Dr. Mohammad Abdullah Al Mumin	Dr. Mohammad Abdullah Al Mumin
Professor	Professor	Professor
Department of Computer Science and Engineering	Department of Computer Science and Engineering	Department of Computer Science and Engineering

Abstract

The neural machine translation (NMT) is rapidly overtaking the state-of-the-art efficiency normally achieved by phrase-based methods (PBMT) in statistical machine translation field, and quickly has become the dominating leading platform in machine translation. On a variety of language pairings, NMT outperformes well-known state-of-the-art PBMT system. We conduct a thorough assessment of statistical *vs.* neural machine translation outputs on the SUPara data, employing high quality post-edits accomplished by expert translator, to see where NMT outperforms SMT in terms of translation quality. Bangla language guidelines with various features, which are considered to be particularly difficult due to morphological and syntactic variations, and where SMT systems generally achieve excellent quality and so offer a powerful alternative for NMT. Our study revealed which language processes, such as words reordering, are best described by neural models, while also highlighting areas where work has to be done, such as the right translation of words.

Keywords: Machine Translation, Statistical Machine Translation, Neural Machine Translation, state-of-the-art

Acknowledgements

First, we express our deepest gratitude to Allah, the Almighty, for pouring His blessings on us throughout our research, enabling us to finish the research successfully. We would want to thank our research supervisor, Professor Mohammad Abdullah Al Mumin, from the bottom of our hearts for giving us the opportunity to conduct research and for giving us with essential advice during this research. His determination, genuineness, and vision have all made an unforgettable imprint on us.He also instructed us how to conduct research and how to present the findings of our analysis as simply as possible. It was a great joy and privilege to work and learn underneath his supervision. We are grateful for what he has provided us. We also would like to thank him for his friendliness, understanding, and amazing sense of humour.

We would like to convey our heartfelt thanks to our mentor, Md Zobaer Hossain, for his guidance and support during this thesis work.

Finally, we would like to express our gratitude to everyone who has helped us in finishing the research project, whether direct or indirect.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the year 2012, the Encyclopedia Britannica declared that they will no longer print any publications. 244 years old publisher has moved out from the printing to publishing online as the technological revolution made access to the internet so easy [1]. This news is nine years old, recent statistics are more astounding. The amount of data produced in the past two years has surpassed all prior human history. The exponential growth of the internet has made this possible. Internet subscribers of Bangladesh have reached 93.702 million users in 2019 at an increased rate of 15-16% per year [2]. This all proves how we have engaged ourselves in these modern technologies.

72.1 percent of internet users prefer to access sites in their native language, according to Common Sense Advisory [3]. However, using human translations on billions of sites or contents is not a viable option. Without any automatic translations system it is not possible to make the internet easier. Machine Translations(MT) is introduced to solve this problem. MT automatically converts the text of one language to another. Statistical machine translations(SMT) and Neural machine translations(NMT) are the two very common approaches that have been used in MT. Both SMT and NMT are data-driven techniques in this case. That is, in order to construct an automated translations system, both strategies need corpora.

1.1 Motivation

The wave of neural models has finally made its way into the SMT area. After a period wherein the NMT proved too analytically and resources expensive to competing with state-of-the-art PBMT.

In comparison to previous paradigms, NMT represents a simplicity. In comparison to rule-based MT, it provides for a more efficient use of human and data resources from a management standpoint, comparable to SMT. The architecture of a massive network structure programmed for end-to-end translation is far more architecturally straightforward than typical MT systems, which incorporate several elements and manufacturing process. The NMT approach, on the other hand, is less translucent than prior models. Indeed, the statistical/data-driven approach is a step ahead from rule-based techniques that clearly influence information, which is nonetheless comprehensible in its internal workings, to something like a sub-symbolic environment in which the analyst has no visibility into the translation process. What are the advantages and disadvantages of NMT versus SMT that we are aware of? What are some of the language phenomena that deep learning translation models can handle so well? To provide answers to these issues and to go beyond the limited information provided by BLEU [3], NIST [4], TER [5]. We compare the two perspectives in this study to throw light mostly on characteristics that differentiate both and prove the substantial differences in quality.

1.2 Goals of the Thesis

The first research of this kind, which included NMT, was conducted by [6]. The primary insight has been that NMT outputs contained fewer lexical and morphological errors, as well as much less word order problems, particularly when it came to verb placement. The study, however, only looked at one language pair and one NMT system. First, we take into account a larger number of NMT and SMT systems as well as a new language pair. This increases the generality of the results while also reinforcing their dependability. Finally, reordering errors were subjected to a really well evaluation on the segmentation of errors by word class. We expand this approach to lexical and morphological errors in this work.

1.3 Structure of the Thesis

We outline this paper as follows:

Chapter 2 In this chapter, we have discussed about background study and literature reviews.

Chapter 3 Previous Work is discussed in this chapter.

Chapter 4 Error evaluation is described here.

Chapter 5 In this chapter, we have discussed about Errors Analysis, Results and Discussion.

Chapter 6 This chapter concludes this report in brief.

Chapter 2

Background Study

2.1 Machine Translation

Machine translation is the use of computer software to translate a text from one language to another without the assistance of a person. Machine translation may be divided into three categories:

Transfer-based MT: This technique is based on the source and target languages, syntactic, morphological, and semantic analyses.

Interlingual MT: The original language is converted into a linguistically representation that is an optional part of the semantics of the text.

Direct MT: Without any intermediary stages in the translation process, the source language is immediately translated into the destination language. The morphological inflections are removed from the source to get the basic form, which is then matched in a bilingual dictionary.

The level of the source language analysis is the key distinction between these three methodologies. The Vauquois Triangle *Figure 2.1* breaks down the procedure into manageable segments. The methodology requires more analysis on the source language side and more generation on the target language side with each level higher. The methodology requires more analysis on the source language side and more generation on the target language side with each level higher. The direct MT, which represents the lowest level in the triangle, is a lexical translation from the source to the target language. Each phrase is turned into an abstract representation at the top level, which symbolizes the Interlingual.

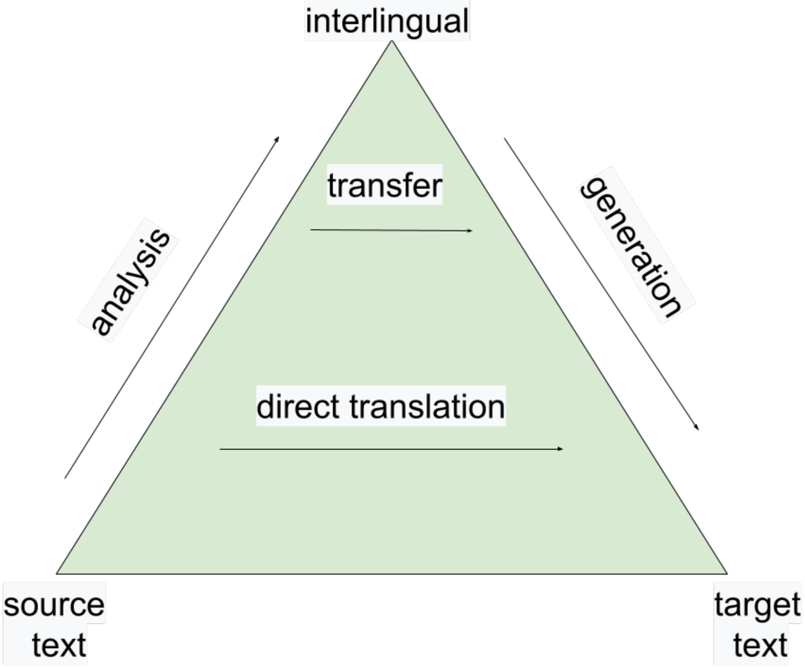

Figure 2.1: Vauquois Triangle [1]

2.2 SMT

In the field of machine translation, statistical machine translation (SMT) is a paradigm in which translations are created on the basis of statistical models, the parameters of which are determined via the examination of bilingual text corpora. The statistical approach to machine translation differs from rule-based and example-based methods to machine translation [7]. The concept of statistical machine translation is derived from the field of information theory. When a document is translated, the probability distribution $p(e|f)$ is used to determine if a string e in the target language is a translation of a string f in the source language.

By examining previous human translations, statistical machine translation (SMT) is a kind of machine translation that learns how to translate (known as bilingual text corpora). In contrast to the Rules-Based Machine Translation (RBMT) approach, which is mostly word-based, most current SMT systems are phrase-based and construct translations utilizing overlap phrases. The aim of phrase-based translation is to get beyond the constraints of word-based translation by translating long sequences of words. Rather than being language phrases, phrase sequences are word sequences that have been found using statistical methods from multilingual text corpora.

2.3 NMT

In the field of machine translation, neural machine translation (NMT) is a methodology that makes use of a convolutional neural network to determine the probability of a sequence of words, generally modeling full sentences in a highly centralized model [8]. NMT differs from phrase-based statistical techniques that rely on subcomponents that are individually created and assembled. NMT (neural machine translation) is not a radical departure from statistical machine translation (SMT). The use of vector representations ("continuous space representations", "embeddings") for words and internal states is the fundamental departure. The models have a simpler structure than phrase-based models.

NMT is a groundbreaking technique to language translation and localization that use deep neural networks and artificial intelligence to develop neural models. With a major transition from SMT to NMT in only three years, NMT has swiftly become the dominant technique to machine translation. Statistical Machine Translation techniques often provide lower-quality translations with less fluency and adequacy than Neural Machine Translation systems.

The memory used by neural machine translation is a fraction of that required by classic Statistical Machine Translation (SMT) models. This NMT methodology varies from traditional translation SMT systems in that all portions of the neural translation model are trained together (end-to-end) to provide the best translation results.

2.4 Lemmatizer

The act of gathering together the variant forms of a word so that they may be studied as an individual entity, designated by the dictionary form or word's lemma, is known as lemmatization (or lemmatization) in linguistics. A Bengali lemmatization method has been designed and tested. Its word sense disambiguation (WSD) efficacy is also explored. One of the primary issues for processing speed of highly inflected language is dealing with the regular morphological changes of the etymological roots that occur in the text. As a result, in order to construct NLP (Natural Language Processing) tools for such language, a lemmatizer is required. Bengali was used as a reference in this research since it is the national language of Bangladesh and the third most common language on Indian sub-continent.

Assume w is a lemmatized surfaces word. On w, the following two operations are defined:

1. The vector in the present environment of w is denoted by CON(w). 2. The part of speech for the word w in the current context is denoted by POS(w).

BenLem requires at least two resources: (i) valid suffix list (ii) dictionary

Bengali Suffix List: The lexicon is used to create a collection of all suffix of length n (n = 1, 2,...) by the number of words in a group and grouping words that share the same suffix becomes the prevalence of the associated suffix. If the frequency of a suffix in the lexicon exceeds a given cut-off criterion, it is referred to as a prospective suffix [9]. The authors revealed that suffixes selected only on their frequency in a lexicon had a strong reliability but a poor stability. The selected set contains the majority of the language's legitimate suffixes, it also contains numerous invalid suffixes. The reason for getting a collection of acceptable suffixes in this manner would be that, while certain isolated lexical resources may include incomplete lists of Bengali suffixes, there is no comprehensive or standard catalogue. Traditional Bengali grammars also don't include a list of all the possible suffixes that need to be considered.

Distance Measure: If w1 and w2 are two words, then *Dist*(w1, w2) represents the string distance between them, and we have chosen the distance measure to create the function, *Dist*, which promotes lengthy matched prefixes while penalizing early mismatches in the following way [10]. Let n+1 be the length of the first mismatch between X and Y and m be the placement of the first discrepancy between X and Y.

$$Dist(X,Y) = \frac{n-m+1}{m} \times (\sum_{i=m}^{n} \frac{1}{2^{i-m}}) \tag{2.1}$$

The rationale for using this distance measure is that morphologically related words in suffixing languages like Bengali usually share a common prefix. So, if *Dist*(w1, w2) is less than a threshold, we may claim that the two words w1 and w2 are morphologically connected.

Available Lemmatizer for Bangla : We have found two lemmatizer in Bangla language. These are BenLem [11] and NeuLem [12]. In our research we have used NeuLem.

2.5 Stemmer

The practice of obtaining the stem or root word from an inflected word is known as stemming. Stemming is the process of reducing many word forms / grammatical to their stem, root, or base form. Information Retrieval Systems [13] frequently utilize stemming.

Readers, researchers, and especially foreigners learning Bangla as a second language need to find stem words to communicate and extract information from documents, newspapers, and other sources.

In the literature, several stemming techniques have been developed, including co-occurrence computing, prefix stripping, dictionary look-up, natural language processing techniques, probabilistic, and longest suffix matching. The majority of techniques are created in English initially and then modified for use in other languages. None of these methods, however, perform well for Indo-Aryan languages (Hindi, Bengali, Gujarati, and Marathi) with a lot of inflection [14].

Bengali, one of the most morphologically diverse languages in the world, with many inflectional and derivational alternative forms of a word, making it difficult to distinguish stem words from inflected terms [15].

Taking into consideration the situation, there are two parts of speech in Bengali: nouns and verbs, both of which include a large number of inflectional suffixes. A few adjectives can also be inflected. The key difficulty here is discovering as well as creating procedures in Bengali to accurately detect stem words for a given set of conjugated terms.

A lot of research articles have been published in the literature that address the stemming challenge for Bengali words [15] [16] [17]. To detect inflections, [18] checks a tokenized word sequentially using a preset origination set, whereas others [15] [19] use hash table matching to remove suffixes. A large amount of database searching being done in every situation. Because of their high time and space complexity, these systems are inappropriate for a wide range of applications. In our research we have used lancasterStemmer [20].

Chapter 3

Previous Work

Machine Translation translates texts of one natural language into texts of another automatically. State of the art MT approaches use parallel corpora as their training data. Finding available Bangla ↔ English parallel corpora as well as measuring quality of them by using common linguistics features, evaluating well-known metrics by following statistical machine translation techniques was our previous work. Thirteen out of fourteen corpora have been founded publicly available. Most of the corpora were constructed using open source domain and preprocessed of their own way.

Since corpora are pre-requisite of MT systems, a large scale corpora will enhance the MT system in Bangla language. However, there are some publicly available corpora on Bangla ↔ English. According to our knowledge, There have been no systematic analyses of the parallel corpora that have been collected so far. An analysis on corpora construction along with their data distribution and quality comparison will create an opportunity for the researchers on this field.

In previous, we reviewed all the available Bangla ↔ English parallel corpora. While reviewing these corpora we explored the system they followed to make their corpora. In addition, we have also taken a look at the data distribution of these corpora. In addition, we employed a state-of-the-art Statistical Machine Translation system to assess their machine translation performance.

3.1 Search Strategy

For developing an effective search strategy, we used a 7-step framework by following the 12-step framework developed by Kable [21]. In addition to serving as a useful tool for recording a systematic review's search strategy, the 7-step framework also serves to guide researchers through the process of identifying and finding relevant literature. Each of the seven stages is presented sequentially in this part in order to improve readability and make it easier for readers to locate specific stages quickly. In 1991, the first attempt was made to develop *Anglabharti* [22], an Bangla ↔ English Machine Translation system as part of the English-to-Indian Languages system. After that, the number of corpora had increased slowly and reached at fourteen. The search did not include any book chapters. The purpose was to discover as well as a thorough analysis of available corpora of row-resource Bangla language developed during the years 1991 to 2021. To study Machine Translation, and Systemic literature review, we limited searches to peer-reviewed journals. As Bangla is a low-resource language, to study Machine translation in Bangla ↔ English, and collect available corpora, we did not limit our search. We studied all articles and collected all datasets we got.

Prof. Dr. Abdullah Al Mumin, the supervisor, helped to design the search keywords. Previously, he had worked in the Machine Translation area, and he was better equipped to create successful search phrases than the author would have been on his own, owing to the author's lack of experience in the area of study. We have used Machine translation as a core text to search by adding *Bangla*, *Bengali*, *row-resources*, *corpus*, *corpora*, *review papers*. The searches were carried out in five different digital libraries, ACM Digital Library [5], IEEE Explore [6], Science Direct Elsevier [7], Springer Link [8] and ACL [9]. In order to complete this work, a conceptual research string comprising the primary keyword of the topic was created.

In ACM library's search engine, we searched all of the strings mentioned in *Table 3.1* by filtering 'Machine Translation' domain in common. If we search without having any domain, engine will show different results and the number of irrelevant papers will be huge because of their searching algorithm.

In SpringerLink, we narrowed the search contents based on the parameters provided on the search page, and only conference papers were counted.

Both ACM and SpringerLink's search engine searches by words in contrast to full strings which is a major problem for getting weird results. For this reason, a huge number of papers come up on the search results, where a maximum of them are not relevant according to the full strings.

Elsevier's searching algorithms seems messy to us because it showed the same results, again and again, despite changing search strings.

	Digital Library					
Term Search	ACM	IEEE	Springer	Elsevier	ACL	Total
Machine Translation	1,129	4201	76,474	283	387	82474
Machine Translation of Low Resources	738	60	18,079	2,656	9	21542
Machine Translation of Bangla	6	32	141	2,656	0	2835
Machine Translation of Bengali	7	21	213	2,656	0	2897
Statistical Machine Translation of parallel corpus	1129	121	1,703	2,656	2	5611
Neural Machine Translation of parallel corpus	1129	30	708	2,656	2	4525
SMT of Bangla parallel corpus	990	1	10	2,656	0	3657
NMT of Bangla parallel corpus	978	0	1	2,656	0	3635
Review paper of Low Resources	1,056	313	57,213	2,669	1	61252
Review paper of Bangla MT	1,052	0	6	2,669	0	3727
Review paper of Parallel Corpora MT	1,085	0	236	2,670	1	3992

Table 3.1: Following the execution of search strings, the number of papers retrieved in each Digital Library.

3.2 Available Corpora

We have found fourteen Bangla ↔ English Parallel corpora by searching through different libraries. Nine of them are found on Opus named *GlobalVoices*, *Gnome*, *JW300*, *KDE4*, *OpenSubtitles*, *QED*, *Tanzil*, *Tatoeba*, *Ubuntu*. Opus attempts to help by collecting new data sets on a big scale in order to offer data for a wide range of languages and topics that are frequently underserved. The Opus project's general aim is to make parallel resources publicly accessible, with a particular focus on low-density languages which is really helpful for the researcher [23]. Others are given below.

SUPara corpus [24] is distributed through the Computer Science and Engineering (CSE) department of Shahjalal University of Science and Technology (SUST).

EMILLE corpus [25] developed through the "Enabling Minority Language Engineering" project, which was undertaken by the universities of Lancaster and Sheffield.

ILMPC (Indic Language Multilingual Parallel Corpus) is introduced on Workshop on Asian Translation (WAT 2018) ([26], [27]).

Pan Treebank Bangla-English parallel corpus ([28], [29]) (PTB) is developed by PAN Localization Project [12].

SIPC (Six Indian Parallel Corpora) is constructed via CrowdSoucring by using Amazonâs Mechanical Turk (MTurk) [30].

All of the founded corpora except *Pan Treebank*, are available free of charge for educational and research purposes, however, the license allows collecting statistical data and making short citations. Maximum corpora are found as *tmx*, *moses* and *txt* formats in different websites. Two of them are not available on internet. So we collected them personally from respected Authors. Some of the corpora have software that uses their own corpus to translate from one language to another.

Here is a discussion about the source of the corpora that we've worked with.

In *JW300* corpus [31], data collected from online website *jw.org*. The vast majority of texts collected from the magazines *Awake!* and *Watchtower*. In this corpus, multilingual articles are mainly translated from the bible to 300 languages.

In *SIPC* corpus [30], they apply an established protocol for using *Amazonâs Mechanical Turk* (MTurk) to collect parallel data to train and evaluate translation systems for six Indian languages. They investigate the relative performance of syntactic translation models and explore the impact of training data quality on the quality of the resulting model. Finally, they release the corpora to the research community under the Creative Commons Attribution-Sharealike 3.0 Unported License.

QED or *AMARA* [32] corpus is an open multilingual collection of subtitles for educational videos and lectures collaboratively transcribed and translated over the AMARA web-based platform. Here maximum contents are collected from *Khan Academy*, *Coursera*, *Udacity*, *TED Talks* and *ELRA* website.

In *OpenSubtitles* corpus [33], The collection is a database dump of the *OpenSubtitles.org* subtitle repository, which contains 3.36 million subtitle files in over 60 languages. This is a very upgraded version of the subtitle gathering that has better sentence alignment and language checks.

SUPara [24] have used texts from some sources that are either publicly available or granted permission from respective copyright holders. They collect data from novel and feature. Some Documents containing Prime Minister's speeches, budget speech, commercial policy, etc are obtained from the government official website. Several essayistic texts are collected from different online newspapers, e.g., Bangladesh Sangbad Sangstha(BSS), BDNews24.com. Many other documents are obtained from websites of several companies like Grameen Phone, Bangladesh Parjatan Corporation and so on. They also generated some documents by mining data from *Wikipedia* and *Banglapedia*.

Tanzil [23] is a collection of Quran translations, and all data are collected from the Quran and their own *website*.

GlobalVoices [23] collected data mainly from *Global Voice* website. *Tatoeba* [23] collected data from *Tatoeba* website.

KDE4, *Ubuntu*, *GNOME* corpora are originally mentioned in OPUS's *website*. We didn't get much idea about its source from [23] paper.

For all available corpora from Opus, website also provides pre-compiled word alignments and phrase tables, bilingual dictionaries, frequency counts, and these files are found there.

3.3 Preprocessed to build corpus

We could not find more details on preprocessing on the corpora. In the following *Table 3.3*, we tried to give a rough idea of the few preprocessing terms corpora that follow. Here, the following steps have been applied as preprocessing on the English and Bangla documents for the available corpora:

Cleaning up Documents - This cleaning up means that the various formats, for example, rtf, doc, and pdf, are converted to plain text files. Tagged files like HTML and PHP files are normalized by deleting tags and then converted to plain text files.

Encoding and Markup - The texts are encoded according to international standards by using UTF8 (Unicode). For Bangla documents, they have used the 'Nikosh' converter to encode all formats into Unicode.

Alignment - The alignment of translated segments with source segments is essential for building parallel corpora. Each file in a sub-directory is aligned separately with its translation to keep alignment errors at a low level.

Tools - Some corpora used various tools for preprocessing the data. In particular, they applied various types of open-source software and free research tools. These tools include sentence splitter,

	Corpus Name				
Preprocessing Terms	GlobalVoices GNOME JW300 Tanzil Ubuntu	OpebSubtitles	QED	Supara	ILMPC
Cleaning up Documents				✓	
Encoding and Markup				✓	
Alignment				✓	
Tools				✓	
Subtitle conversion		✓			
Sentence segmentation	✓	✓	✓		✓
Tokenization	✓	✓	✓		✓
Correction of OCR	✓	✓			
Spelling errors	✓	✓			
Inclusion of meta-data		✓			
Training			✓		✓

Table 3.2: Statistics overview of Corpora of how they preprocessed themselves.

word histogram generator, Unicode converter, etc.

Subtitle conversion - Only the OpenSubtitles corpus follows these preprocessing words, as seen in *Table 3.2*. No specific encoding format is required for the subtitles submitted by its users, and OpenSubtitles does not impose any such requirement. In order to avoid this, it is necessary to find the most probable encoding for the file using a variety of heuristics. When there are numerous acceptable alternative encodings available, the 'chardet' library is used to select the most probable encoding based on the file content [34].

Sentence segmentation and tokenization - In *Table 3.2*, we can see most of the corpora follow these preprocessing terms except the Supara corpus.

Correction of OCR and spelling errors - Because a large percentage of subtitles in our dataset are automatically generated from video streams using Optical Character Recognition (OCR), there are a significant number of OCR errors. In *Table 3.3*, we can see most of the corpora follow these preprocessing terms except the Supara corpus, the QED corpus, the ILMPC corpus.

Inclusion of meta-data - This preprocessing phase's goal is to create the meta-data that will be connected with each individual subtitle.

	En → Bn			Bn → En		
Corpus Name	BLEU ↑	NIST ↑	TER ↓	BLEU ↑	NIST ↑	TER ↓
EMILLE	0.37	1.14	113.12	1.26	1.70	93.35
Gnome	0.77	1.15	113.56	0.82	1.48	93.32
GlobalVoices	7.35	3.35	86.04	8.66	4.13	82.40
ILMPC	2.90	2.24	98.49	4.50	3.08	88.31
KDE4	0.67	1.08	117.09	1.31	1.55	94.46
OpenSubtitles	2.54	2.15	98.78	4.44	3.11	88.20
QED	0.00	0.68	116.09	0.36	1.12	95.98
SIPC	7.36	3.29	85.60	7.02	3.56	87.07
SUPara	**8.87**	**3.54**	**85.24**	**9.88**	**4.49**	**79.45**
Tatoeba	0.28	0.68	118.39	0.38	0.81	95.43
Tanzil	0.00	0.84	109.87	1.16	1.33	96.23
Ubuntu	0.00	0.84	119.05	0.68	1.17	94.61

Table 3.3: Evaluation scores of all corpora using *sumono.5-gram.blm.bn* as language model for English → Bangla and *europarl.5-gram.blm.en* as language model for Bangla → English translation (Sorted Alphabetically) [SUParadev for tuning and **SIPCtestset** for evaluating]

3.4 Distribution

JW300 is the largest having huge coverage of languages, almost 380 languages mentioned in *Table 3.2*. Tatoeba covers the second most languages. But Bangla ↔ English parallel corpus is tiny having 5,120 sentences mentioned in *Table 3.5*. Though GlobalVoices has the lowest language coverage among other corpora of Opus, Bangla ↔ English side is pretty huge having 137,620 sentences. The number of language coverages of other Opus corpora is given in *Table 3.4*. All available corpora of Opus are bidirectional. Which means, Bangla to English or English to Bangla translation both are possible.

ILMPC corpus covers eight languages. These are Bangla, Hindi, Malayalam, Tamil, Telegu, Sinhalese, Urdu and English. This corpus is used for the pilot as well as multilingual English-Indic or Indic-English Languages sub-tasks. It is a collection of 7 bilingual parallel corpora of varying sizes, one for each Indic language and English. The parallel corpora are also accompanied by monolingual corpora from the same domain.

SUPara corpus is an Bangla↔English parallel corpus consisting of more than 0.45M words in either languages, which is the largest among freely released corpus of its kind.

EMILLE corpus consists of a series of monolingual corpora for fourteen South Asian Languages and a parallel corpus of English and five of these languages.

SIPC is a collection of parallel corpora between English and six languages from the Indian subcontinent: Bangla, Hindi, Malayalam, Tamil, Telugu, and Urdu.

There are 152,939 movies or TV episodes covered by OpenSubtitles (as determined by IMDb identifier). 8% of the IMDb identifiers are associated with subtitles in at at least 20 languages, 28% with at least 10 languages, 44% with at least 5 languages, 70% with at least 2 languages. However, the reason for the large OpenSubtitles files is because, in the event of numerous CDs, each movie or TV show may contain numerous files. QED or AMARA corpus covers 44,620 online educational videos, TV shows such as Khan Academy, TED Talks, Udacity, Coursera and generates around 271,558 files from their subtitle files. JW300 generates files from Bibel, New Testament, Awake! and Watchtower magazines by segmenting each chapter or article into a reasonable size. Tanzil builds 80 files by splitting the Holy Quran into 80 sections. SUPara corpus contains altogether 80 document pairs that consist of literature, journalistic, instructive, administrative, external communication. Other corpora mentioned in *Table 3.4* follow the same rule to generate files.

Sentence fragment is a punctuated word, phrase, or dependent clause that lacks a subject, verb, or both. While sentence fragments may be used for impact in certain kinds of writing, they are seldom utilized in academic or professional writing. Some of the corpora especially Opus used fragments of sentences. Others (such as ILMPC, SUPara) used full sentences, instead of using a fragment. We found two different corpora of JW300, shown in *Table 3.5*. Every corpus except JW300$_1$ and JW300$_2$ [13] have same number of sentences or fragments of sentence. These two corpora are not even properly aligned with each other which is so much important for translation. For example, Bangla document's fifth line is not aligned with the English document's fifth. There exists excessive new lines (empty lines) instead of proper alignment.

OpenSubtitles has the maximum number of sentence fragments around 3.35G because of the availability of online subtitles' sources. Also, Bangla ↔ English side of OpenSubtitles has the largest number of sentences. Overall count of sentence fragments for EMILLE and ILMPC are not mentioned in the dataset's description. SUPara has only 0.02M sentences because it covers only two languages. But SUPara is one of the finest datasets of it's kind. Count of sentences or fragments of other corpora is mentioned in *Table 3.5*.

Token is the smallest unit that each corpus divides to. Typically each word form and punctuation (comma, dot) is a separate token. Therefore, corpora contain more tokens than words. Spaces between words are not tokens. But in *Table 3.3*, the term “Token” actually refers to the total number of words in the corpus, of course excluding punctuations. OpenSubtitles has the largest number of tokens nearly 22.1G because of its data collection policies. SUPara has 0.45Million tokens because it covers only two languages. Others are mentioned in *Table 3.4*.

“Words” which is mentioned in *Table 3.5*, denotes the total number of barely Bangla words in Bangla side and English words in English side corpus individually. Some corpora (such as Gnome, Ubuntu, QED) have Bangla sentences having a lot of English words. We excluded English words from them and counted only Bangla words. $JW300_2$ [13] has the highest number of words in both English and Bangla side. QED or *AMARA* corpus [32] has 71,695 English words in contrast to 245,548 Bangla words. This means, almost every sentence of Bangla corpus has more words than the aligned English sentence. Translated English sentence is incomplete for almost every sentences of Bangla. That’s why QED has the highest “average words per sentence” in Bangla side which is 120.190 whereas aligned English side has an average 35.093 words per sentence. Number of “Unique Words” of each corpus are given in *Table 3.5*. First, we tokenized them into merely Bangla words for Bangla side and English words for English side and then used set to find out unique words. GlobalVoices has the highest unique words for both in Bangla and English side.

Lexical diversity refers to the ratio of various distinct word stems (types) to the total number of words, which is one element of “lexical richness”. It refers to the range of different words used in a text, with a greater range indicating a higher diversity. Details of lexical diversity of available corpora are shown in *Figure 3.1*. Big score of *Tanzil* refers that on average each vocabulary item appears very frequently for both English and Bangla sides. Ubuntu got the lowest score which means items appear very rarely on sentences for both English and Bangla side.

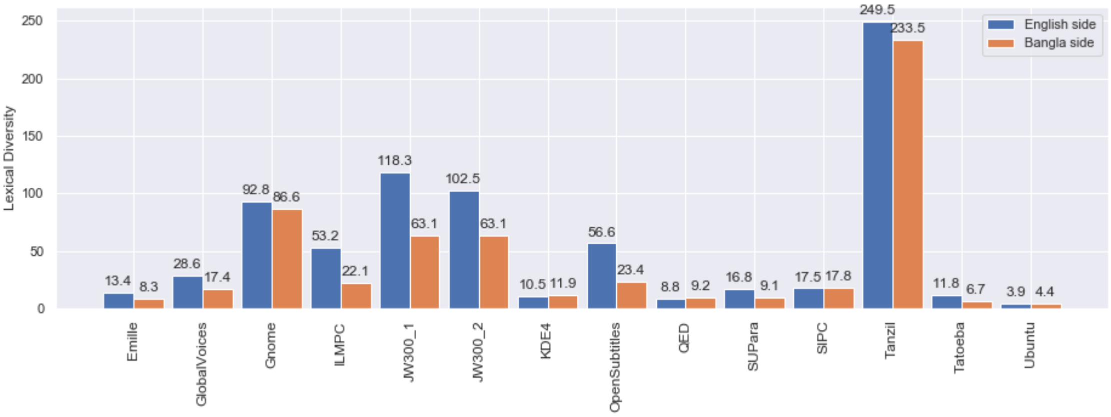

Figure 3.1: Lexical Diversity of available corpora

Corpus Name	Languages	Files	Sentence Fragments	Tokens
Emille	14			67M
GlobalVoices	41	224,096	4.93M	88.71M
Gnome	187	113,344	58.12M	267.27M
ILMPC	8	3,200		4.64M
JW300	**380**	1,285,939	105.11M	1.95G
KDE4	92	75,535	8.89M	60.75M
OpenSubtitles	62	**3,735,070**	**3.35G**	**22.10G**
QED	225	271,558	30.93M	371.76M
SUPara	2	80	0.02M	0.45M
SIPC	6		0.15M	1.6M
Tanzil	42	105	1.01M	22.33M
Tatoeba	309	309	7.82M	57.55M
Ubuntu	244	30,959	7.73M	29.84M

Table 3.4: *Bird's-eye-view of founded Parallel Corpora (Sorted Alphabetically)*

	Sentences	Words		Unique Words		Average Words per Sentence	
Corpus Name	En or Bn	En	Bn	En	Bn	En	Bn
Emille	6,375	89,027	90,062	6,636	10,816	13.965	14.127
Global Voices	137,620	2,536,451	2,269,045	**88,602**	**130,606**	18.4308	16.4877
Gnome	132,481	637,363	619,182	6,869	7,147	4.811	4.674
ILMPC	337,428	2,260,636	1,840,722	42,510	83,418	6.710	5.455
JW300$_1$	366,972	5,180,241	5,074,551	43,781	80,447	14.116	13.828
JW300$_2$	370,948	**5,890,630**	**5,083,847**	57,461	80,575	2.332	13.705
KDE4	36,381	149,273	129,159	14,174	10,869	4.103	3.550
OpenSubtitles	**413,602**	2,401,653	1,974,181	42,432	84,343	5.807	4.773
QED	2,043	71,695	245,548	8,167	26,674	**35.093**	**120.190**
SUPara	21,158	244,539	202,866	14,571	22,456	11.56	9.59
SIPC	20,788	290,972	240,077	16,594	13,501	13.997	11.549
Tanzil	187,052	4,391,125	4,185,894	17,600	17,925	23.475	22.378
Tatoeba	5,120	24,656	22,321	2,082	3,345	4.816	4.360
Ubuntu	5,634	21,791	17,900	5,619	4,053	3.868	3.177

Table 3.5: *Statistics Analysis of Available English ↔ Bangla Parallel Corpora (Sorted Alphabetically)*

Chapter 4

Error Evaluation

An automatic evaluation produces a single numerical score that assesses the machine translation's performance, but it provides no information regarding the system's flaws. A manual evaluation enables system developers to assess the quality of their MT and evaluate the system's output errors.

As shown in [2], errors can be categorised using a hierarchical structure 2.2. The errors are classified into five major categories: "missing words", "incorrect words", "word order", "punctuation" and "unknown words". When a word is missing, a "missing word" error can occur.

The next type error category is "word order," which is divided into two categories: "word-level" and "phrase-level" reordering, with local and long-range reordering within each of these categories. Individual words should be moved to construct a correct sentence if a "word order reordering" issue occurs. A block of consecutive words should be reordered in the event of "level reordering". The distinction between local and long range reordering is that in the former, words should be reordered in a local chunk, whilst in the latter, words should be reordered into a different context.

The "incorrect words" category is the third type of error. It occurs when the algorithm is unable to find an accurate translation of a term. There are five subcategories: "sense," which refers to meaningless sentences, "incorrect form," which refers to incorrect word forms, "extra words" in the target sentence, "style," which refers to poor word choices, and "idioms," which refers to idiomatic expressions that the system attempts to translate as normal text.

Last but not least, there are errors involving "unknown words," which are divided into "really

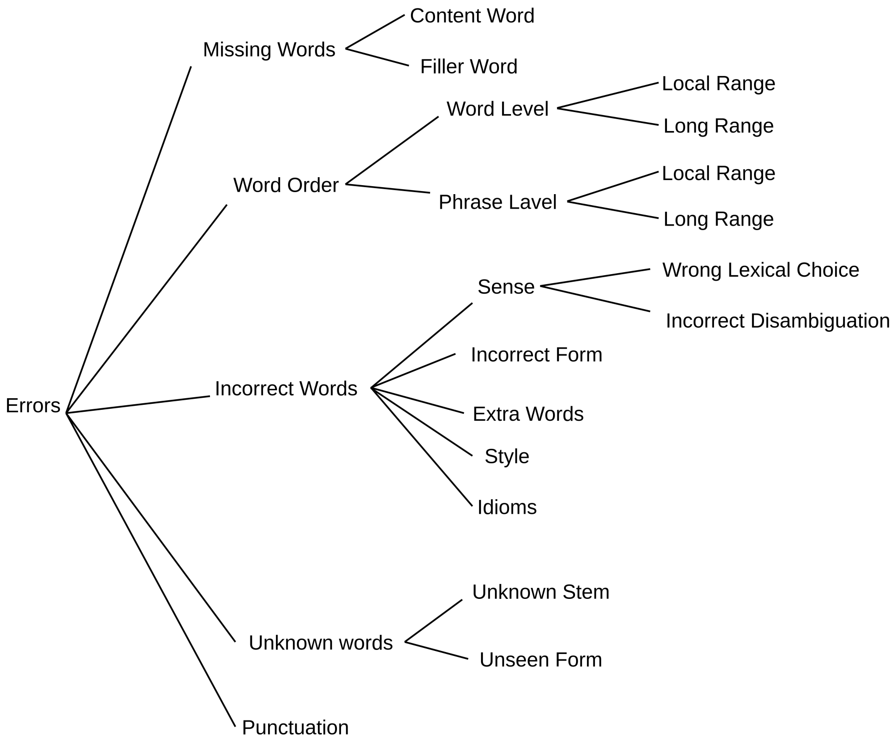

Figure 4.1: Classification of translation errors [2]

unknown words" and "unseen words."

The last sort of mistake is "punctuation," which denotes minor errors in the current machine translation output.

4.1 Evaluation Metrics

By using Statistical Machine Translation System, we evaluated translation results by three metrics: BLEU, NIST, TER. BLEU scores were calculated using *multi-bleu.perl* in the Moses toolkit [35]. NIST scores were calculated using *mteval-v13a.pl* in the Moses toolkit. To get the proper scores of NIST, you must have to wrap translated file into sgm format by using *wrap-xml.perl*. The amount of editing that a person would have to do to update a system output so that it perfectly matches a reference translation is measured by TER (Translated Edit Rate). TER scores were calculated using *tercom.version.jar* [11] which use JDK and JRE to run.

In this paper, we reviewed all available corpora of Bangla ↔ English we had found. The following section is devoted to data sources, distribution, evaluation. Thereafter, we discuss provided tools and, finally, we present our plans for future work.

4.2 Automatic Evaluation

Human assessment is one of many approaches for evaluating machine translation. It is vast, but it is also quite costly, requires a large number of human resources, and may take months to complete. As a result, an automated assessment is required. It should be low-cost, rapid, consistent, and have a good correlation with human assessment. BLEU, NIST, TER, and other automated assessment techniques for machine translation have been used. The Bleu technique is the most popular.

4.2.1 BLEU

Bleu is an acronym for "Bilingual Evaluation Understudy" [3]. Using this algorithm, automatically evaluates the quality of a machine translation on a constant basis. The hypothesis translation is compared to one or more reference translations in this form of comparison. A hypothesis translation, also known as a candidate translation, is a translation that must be examined before it is accepted. In most cases, the reference translation is the accurate translation used to evaluate and contrast the hypothesis. When the candidate translation has a large number of words and phrases that are identical to those in the reference translations, translation is considered to be satisfactory. The following is the formula used by the Bleu to calculate the score:

$$Bleu = BP \times exp(\sum_{n=1}^{4} w_n \times logp_n) \tag{4.1}$$

The Bleu-Score is calculated using an adjusted n-gram accuracy. To arrive at the final result, multiply the number of words from the system translation that occur in any one reference translation by the total number of words in the candidate translation. The program determines the number of 1-, 2-, 3-, and 4-grams in each text using a mathematical formula. The BP (Brevity Penalty) is used to account for the differences between the hypothesis and reference translations when the hypotheses translation is lower. The Bleu-Score can be a positive value from 0 to 1. This score indicates the functional similarity between the method and the reference. When the result is around 1, the machine translation contains a big number of sentences that are comparable to those in the

EnBn			
	BLEU ↑	NIST ↑	TER ↓
SMT	0.1502	5.0955	0.7259
NMT	0.1675	5.1453	0.6630

Table 4.1: Comparison on SMT and NMT

reference translation, indicating that the translation is excellent. In *Table 4.1* and *Table 4.2*, we compare SMT and NMT using NIST, BLEU and TER score.

4.2.2 NIST

Other recent endeavors, such as investigations of the association between various human evaluations and various automated measurements, as done in recent WMT workshops [36], [37], demonstrate the interest in enhancing MT metrology. The NIST [4] Measurements for Machine Translation Challenge (MetricsMATR) has the unique purpose of focusing only on MT metrology research, bringing together many research initiatives in the area of MT metrology, and assisting in the creation of automated metrics. Researchers may discuss ideas on MetricsMATR [4].

To effectively capture the strengths and drawbacks of MT measures, they must be analyzed across vast and diverse data sets. For example, one would wish to compare the relative performance of measures depending on certain factors like the source language, the kind of MT system (statistical, rule-based, or hybrid), and the data genre. MetricsMATR makes use of a variety of data sets collected by NIST-coordinated MT assessments. Each data collection includes one or more reference translations, one or more machine translations, and one or more human evaluation types.

The analysis for MetricsMATR was based on the principle that the closest an automated measure resembles human assessors, the better the metric. As a result, human evaluation is critical in this task. Human evaluations of many forms are available and will be used to compare metrics scores against. Finding the optimal approach to conduct human evaluations is a huge scientific problem in and of itself. One problem is achieving appropriate intra- and inter-annotator agreement; another is devising evaluations that can be completed in a fair amount of time and effort. The IWSLT 2006 evaluation campaign [38], the WMT-07 [36] and WMT-08 [37] workshops, and the NIST OpenMT 2009 assessment are all recent projects that investigated intra- and/or inter-annotator agreement and revealed a need for improvement. WMT-08 focused on enhancing human evaluation techniques by enhancing intra- and inter-annotator agreement and minimizing assessment time (by assessing at the sub-sentential element level, as opposed to MetricsMATR's approach of assessing at the sentence level).

4.2.3 TER

It is a statistic for evaluating the output of a machine translation program that is automatically generated [5]. It counts the number of modifications that a person would have to do in order to convert a system output into one of the reference values. Replacements of words, deletions, changes of a word sequence, and insertions are all examples of changes that may be required. The term TER is defined as follows:

$$TER = \frac{number \quad of \quad edits \quad needed}{average\ number\ of\ reference\ words}$$

4.2.4 HTER

We utilize HTER to assess the distance between the MT output and its post-edited version in order to comment each sentence for translation quality [5]. HTER calculates how much editing a human would have to do to modify the MT output to make it a decent translation. As a result, the human post-edited version is used as the reference translation in this case.

4.3 Standard word error rates (overview)

The Levenshtein distance [39], the smallest number of substitutions, deletions, and insertions required to change the generated text hyp into the reference text refâis used to calculate the word error rate (WER). The WER has a flaw in that it does not enable reordering of words, despite the fact that the hypothesis word order may differ from the reference's, even if the translation is correct. The position independent word error rate (PER) compares the words in the two phrases without taking the word order into account to solve this problem. The PER is always equal to or less than the WER. The PER, on the other hand, has a flaw in that the word order can be crucial in some circumstances. As a result, calculating both word mistake rates is the optimum solution.

Calculation of WER :

$$WER = \frac{1}{N^{*}_{ref}} \; (\sum_{k=1}^{k} min \; d_L(ref_{k,r}, hyp_k) \tag{4.2}$$

The Levenshtein distance between the reference sentence ref k,r and the hypothesis sentence hyp k is dL (ref k,r, hyp k). WER is calculated with the help of a dynamic programming algorithm.

Calculation of PER : The numbers n(e, hyp k) and n(e, ref k,r) of a word e in the hypothesis sentence hyp k and the reference sentence ref k,r, respectively, can be used to determine the PER:

$$PER = \frac{1}{N^{*}_{ref}} \; (\sum_{k=1}^{k} min \; d_{PER}(ref_{k,r}, hyp_k) \tag{4.3}$$

Chapter 5

Errors Analysis, Results and Discussion

Now we will look at the types of language errors differentiate NMT vs SMT. We focus on the three types of errors: (i) morphological errors, (ii) lexical errors, and (iii) word order errors. When it comes to lexical errors, a number of current taxonomies differentiate between additional words, and incorrect lexical choice, missing words. The difficulties of distinguishing between such three categories [40], we opt to use coarse-grained linguistic error categorization that includes all of them as lexical errors [41]. We use TER calculation for error analysis because we believe that, because this approach is particularly effective for detecting MT errors since the targeted translation is generated by post-editing the given MT output. We recognize that translation objectivity remains a problem. However, we choose to focus on what a human implicitly marked as a translation error in this finer-grained study, rather than the system's total performance. This is especially true in our assessment methodology, which aims to evaluate the NMT and SMT methods in terms of individual errors made. To classify the errors, we initially lemmatize all MT outputs including corresponding post-edits. The lemmatized outputs are then compared to the lemmatized post-edits that correspond to them. We can statistically analyze the three error types indicated below because we keep record of the associated actual word for each lemma.

Lexical errors: The numbers of lemma deletion, insertion, and replacement instances is added together to decide the quantity of lexical errors. To separate morphological errors from genuine lexical choice issues, it is important to consider TER operations at the lemma level.

Morphology errors: When the post-edit lemma and the MT output lemma are the same, but related word forms are different, morphology errors occur.

Reordering errors: To identify shift operations, the TER metric is used at the lemmas level to calculate reordering errors. WER, but not PER, takes into consideration differences in word order in the hypothesis in relation to the reference. As a result, a reordering error is defined as a term that appears in both the reference and the hypothesis but is tagged as a WER error.

Note that there are times when reordering and morphological error occur on same word, morphology and reordering categories might overlap. After a shift, when the MT lemma match with the post-edit lemma but the word forms are unique, this occurs. The prevalence of error categories is given for each language direction and MT method is shown in *Table 5.2*. As a first general finding, we can see that NMT reduces total error significantly when compared to SMT. We see the same general trend in both languages when it comes to each error category: on lexical errors (EnBn and BnEn), the lowest but still substantial reduction is obtained, while on reordering errors, the largest reduction is obtained, especially when a morphological error is occurring in addition to a reordering error (EnBn and BnEn). It is worth noticing, both generally and for each error category, error reductions are larger in the EnBn direction.

Lexical errors are the most common in both language direction and approaches when it comes to lexical errors. The proportion of lexical errors in the total amount of error is higher in NMT than in SMT: *vs.* for EnBn, *vs.* for BnEn, showing that lexical choice is much more difficult for NMT. SMT systems produce the same amount of lexical errors in both languages (38.79 vs. 42.35), but NMT systems perform worse in Bangla than in English (30.19 vs. 43.73). As a result, the neural method ensures a higher decrease on Bangla (8.60) than on English (1.38) for this type of error.

In terms of morphology, NMT systems produce almost the same amount of errors in both languages, but SMT systems are slightly weaker in Bangla than in English. This indicates that the neural method provides a higher decrease in morphological errors on Bangla than on English.

Finally, it is well recognized that the issue of word reordering affects EnBn more than BnEn. In our trial, the SMT and NMT methods create on average and errors on Bangla, respectively, which are reduced to and errors on English. NMT produces the greatest error reduction in both languages, in Bangla and in English. Despite the fact that reordering errors make up a tiny portion of the total, this decrease is significant since reordering errors are generally inconvenient for the user.

EnBn			
	SMT (%)	NMT (%)	Δ(%)
Matched lemma	57.05	53.86	3.19
Average new lemma per sentence	41.75	41.32	0.43
Average missing lemma per sentence	42.95	46.14	3.19
Overall lexical error	42.35	43.73	1.38
BnEn			
Matched lemma	62.74	65.78	3.04
Average new lemma per sentence	40.33	26.16	14.17
Average missing lemma per sentence	37.26	34.22	3.04
Overall lexical error	38.79	30.19	8.60

Table 5.1: Distribution of lexical error per language direction

5.0.1 Lexical errors:

The distribution of lexical errors by missing words and new words is shown in *Table 5.1*. To begin with, we can observe that the NMT error reductions are quite widely divided across missing words and new words. The amount of error in this categories are small in absolute terms, their negative influence on adequacy is usually significant. This difficulty can be explained in part of the well-known problem on uncommon words translation in NMT, has a variety of solution [42]. We also found that NMT has no effect on lexical errors that include numbers. These error iof the format incompatibilities, such as words vs. digits, which are frequent in NMT and SMT, according to manual inspection.

5.0.2 Morphology errors:

The breakdown of morphological errors is seen in *Table 5.2*. We can observe that NMT generates less morphological mistakes in both languages and across most word classes. We may infer that NMT does a better job of capturing agreement phenomena than SMT. This supports

EnBn			
	SMT (%)	NMT (%)	Δ(%)
Morphology errors	14.64	11.73	2.91
Reordering errors	21.35	12.67	8.68
Morphology + Reordering errors	4.69	2.63	2.06
BnEn			
Morphology errors	5.40	5.38	0.02
Reordering errors	29.38	20.16	9.22
Morphology + Reordering errors	2.17	1.54	0.63

Table 5.2: Morphology and Reordering error are apply to the same word

[43] previous findings, in which an EnGe NMT model achieve near-human correctness on two morphological cooperation tests. NMT, has no effect on morphological errors in Bangla or English words. The fact that these errors are outliers rather than fascinating linguistic phenomena implies that they are outliers.

5.0.3 Reordering errors:

The breakdown of reordering errors is seen in *Table 5.2*. This is the error category that NMT reduces the greatest, as previously stated. Reordering errors, in contrast to lexical and morphological errors, are decreased for all word classes, but to variable degrees. The majority of observations in this category are language-specific. When translating into Bangla, reordering is especially challenging since the location of words in this language fluctuates depending on the sentence type. Despite being trained on raw parallel data without any syntactic annotation or explicit modeling of word reordering, NMT decreases word order errors by an astonishing amount, even when it is syntax-informed. This finding shows that the recurrent neural language model at the heart of the NMT architecture is very good at producing well-formed sentences, even in languages like Bangla, where word order is less predictable.

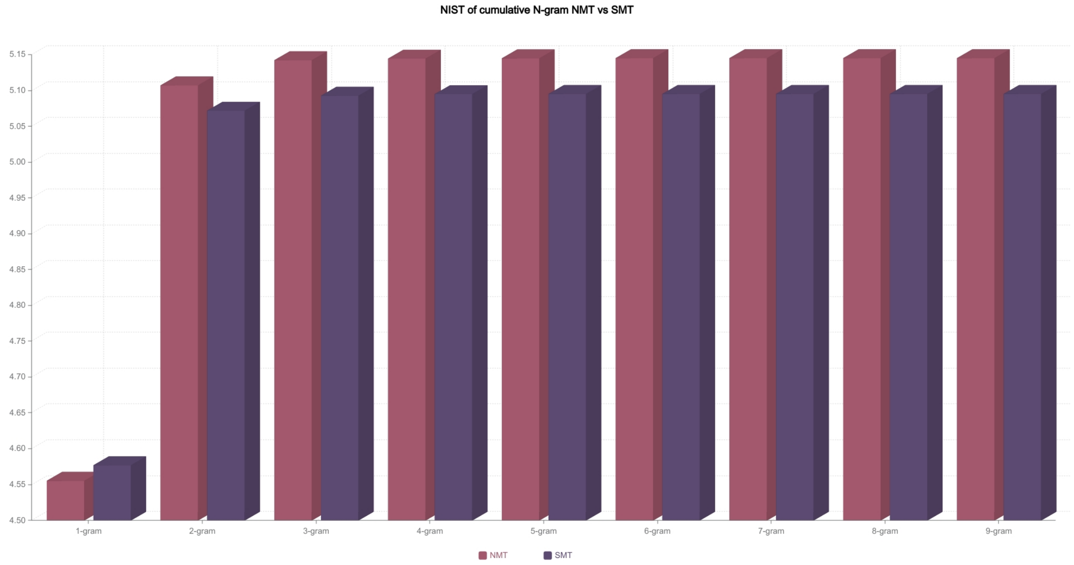

Figure 5.1: NIST N-gram scoring on NMT and SMT for EnBn SUPara Data

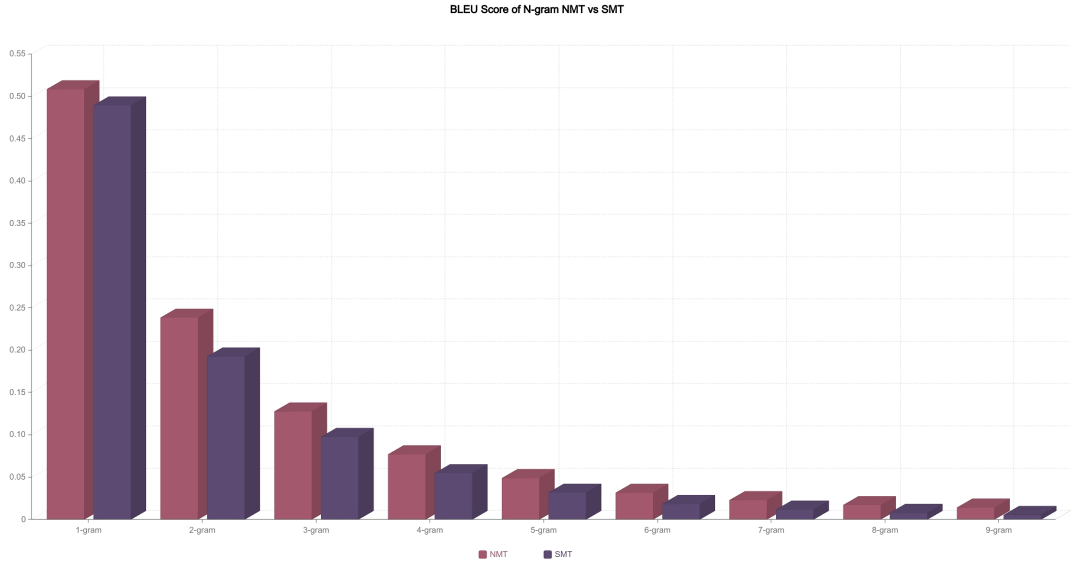

Figure 5.2: BLEU N-gram scoring on NMT and SMT for EnBn SUPara Data

Chapter 6

Conclusion

We looked at the results of many state-of-the-art machine translation systems which took part in Bangla ↔ English competition. Unlike the evaluation model generally used in MT research area, which relies on fair reference translations, our research relies on quality post-edits of MT results, which allow us to analyze systems based on accurate metrics of post-editing effort and different types of translation errors.

The findings show that NMT has considerably advanced the state of the art, just not in language pair with deep morphology predictions and considerable words reordering, and also in a lengthy pair in which the SMT method produced excellent results. To summarize our findings, we discovered: (i) In comparison to SMT systems, NMT provides outputs that significantly reduce overall post-edit work. (ii) NMT beats SMT on all sentence length, with no loss of performance greater than PBMT as source lengths increase. iii) In NMT, the number of MT results with little error is significantly larger than in SMT. (iv) NMT makes far fewer errors than SMT; (v) NMT output had very little lexical, morphological, and reordering errors than SMT output.

References

[1] B. J. Dorr, E. H. Hovy, and L. S. Levin, “Machine translation: Interlingual methods,” 2004.

[2] D. Vilar, J. Xu, D. Luis Fernando, and H. Ney, “Error analysis of statistical machine translation output.” in *LREC*. Citeseer, 2006, pp. 697–702.

[3] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[4] M. Przybocki, K. Peterson, S. Bronsart, and G. Sanders, “The nist 2008 metrics for machine translation challengeâoverview, methodology, metrics, and results,” *Machine Translation*, vol. 23, no. 2-3, pp. 71–103, 2009.

[5] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, “A study of translation edit rate with targeted human annotation,” in *Proceedings of association for machine translation in the Americas*, vol. 200, no. 6, 2006.

[6] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, “Neural versus phrase-based machine translation quality: a case study,” *arXiv preprint arXiv:1608.04631*, 2016.

[7] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[8] K. Wołk and K. Marasek, “Neural-based machine translation for medical text domain. based on european medicines agency leaflet texts,” *Procedia Computer Science*, vol. 64, pp. 2–9, 2015.

[9] J. H. Paik, M. Mitra, S. K. Parui, and K. Järvelin, "Gras: An effective and efficient stemming algorithm for information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 29, no. 4, pp. 1–24, 2011.

[10] M. Prasenjit, M. Mandar, K. S. K. Parui, K. Gobinda, M. Pabitra, and D. Kalyankumar, "Yass: Yet another suffix stripper," *ACM Transactions on Information Systems*, vol. 25, no. 4, pp. 18–38, 2007.

[11] A. Chakrabarty, O. A. Pandit, and U. Garain, "Context sensitive lemmatization using two successive bidirectional gated recurrent networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1481–1491.

[12] A. Chakrabarty, A. Chaturvedi, and U. Garain, "A neural lemmatizer for bengali," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 2558–2561.

[13] D. Bijal and S. Sanket, "Overview of stemming algorithms for indian and non-indian languages," *arXiv preprint arXiv:1404.2878*, 2014.

[14] N. Saharia, K. M. Konwar, U. Sharma, and J. K. Kalita, "An improved stemming approach using hmm for a highly inflectional language," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2013, pp. 164–173.

[15] S. Das and P. Mitra, "A rule-based approach of stemming for inflectional and derivational words in bengali," in *IEEE Technology Students' Symposium*. IEEE, 2011, pp. 134–136.

[16] S. Dasgupta and V. Ng, "Unsupervised morphological parsing of bengali," *Language Resources and Evaluation*, vol. 40, no. 3, pp. 311–330, 2006.

[17] M. Islam, "Research on bangla language processing in bangladesh: progress and challenges," in *8th international language & development conference*, 2009, pp. 23–25.

[18] S. Sarkar and S. Bandyopadhyay, "Design of a rule-based stemmer for natural language text in bengali," in *Proceedings of the IJCNLP-08 workshop on NLP for Less Privileged Languages*, 2008.

[19] M. Islam, M. Uddin, M. Khan *et al.*, “A light weight stemmer for bengali and its use in spelling checker,” 2007.

[20] C. D. Paice, “Another stemmer,” in *ACM Sigir Forum*, vol. 24, no. 3. ACM New York, NY, USA, 1990, pp. 56–61.

[21] A. K. Kable, J. Pich, and S. E. Maslin-Prothero, “A structured approach to documenting a search strategy for publication: A 12 step guideline for authors,” *Nurse education today*, vol. 32, no. 8, pp. 878–886, 2012.

[22] R. Sinha, K. Sivaraman, A. Agrawal, R. Jain, R. Srivastava, and A. Jain, “Anglabharti: a multilingual machine aided translation project on translation from english to indian languages,” in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 2. IEEE, 1995, pp. 1609–1614.

[23] J. Tiedemann, “Parallel data, tools and interfaces in opus,” in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*, N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.

[24] M. A. Al Mumin, A. A. M. Shoeb, M. R. Selim, and M. Z. Iqbal, “Supara: a balanced english-bengali parallel corpus,” *SUST Journal of Science and Technology*, pp. 46–51, 2012.

[25] P. Baker, A. Hardie, T. McEnery, H. Cunningham, and R. J. Gaizauskas, “Emille, a 67-million word corpus of indic languages: Data collection, mark-up and harmonisation.” in *LREC*, 2002.

[26] T. Nakazawa, N. Doi, S. Higashiyama, C. Ding, R. Dabre, H. Mino, I. Goto, W. P. Pa, A. Kunchukuttan, S. Parida *et al.*, “Overview of the 6th workshop on asian translation,” in *Proceedings of the 6th Workshop on Asian Translation*, 2019, pp. 1–35.

[27] T. Banerjee, A. Kunchukuttan, and P. Bhattacharyya, “Multilingual indian language translation system at wat 2018: Many-to-one phrase-based smt,” in *Proceedings of the 5th Workshop on Asian Translation (WAT2018)*, 2018.

[28] M. A. Hasan, F. Alam, S. A. Chowdhury, and N. Khan, “Neural vs statistical machine translation: Revisiting the bangla-english language pair,” in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2019, pp. 1–5.

[29] ——, “Neural machine translation for the bangla-english language pair,” in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2019, pp. 1–6.

[30] M. Post, C. Callison-Burch, and M. Osborne, “Constructing parallel corpora for six indian languages via crowdsourcing,” in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2012, pp. 401–409.

[31] Ž. Agić and I. Vulić, “JW300: A wide-coverage parallel corpus for low-resource languages,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3204–3210. [Online]. Available: https://www.aclweb.org/anthology/P19-1310

[32] A. Abdelali, F. Guzman, H. Sajjad, and S. Vogel, “The amara corpus: Building parallel language resources for the educational domain.” in *LREC*, vol. 14, 2014, pp. 1044–1054.

[33] P. Lison and J. Tiedemann, “Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles,” 2016.

[34] S. Li and K. Momoi, “A composite approach to language/encoding detection,” in *Proc. 19th International Unicode Conference*, 2001, pp. 1–14.

[35] P. Koehn, M. Federico, W. Shen, N. Bertoldi, O. Bojar, C. Callison-Burch, B. Cowan, C. Dyer, H. Hoang, R. Zens *et al.*, “Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding,” in *Final Report of the Johns Hopkins 2006 Summer Workshop*, 2007.

[36] C. Callison-Burch, C. S. Fordyce, P. Koehn, C. Monz, and J. Schroeder, “(meta-) evaluation of machine translation,” in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, pp. 136–158.

[37] ——, “Further meta-evaluation of machine translation,” in *Proceedings of the third workshop on statistical machine translation*, 2008, pp. 70–106.

[38] R. Zhang, H. Yamamoto, M. Paul, H. Okuma, K. Yasuda, Y. Lepage, E. Denoual, D. Mochihashi, A. Finch, and E. Sumita, “The nict-atr statistical machine translation system for the iwslt 2006 evaluation,” in *International Workshop on Spoken Language Translation (IWSLT) 2006*, 2006.

[39] V. I. Levenshtein *et al.*, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[40] M. Popović and H. Ney, “Towards automatic error analysis of machine translation output,” *Computational Linguistics*, vol. 37, no. 4, pp. 657–688, 2011.

[41] S. Jean, O. Firat, K. Cho, R. Memisevic, and Y. Bengio, “Montreal neural machine translation systems for wmtâ15,” in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 2015, pp. 134–140.

[42] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.

[43] R. Sennrich, “How grammatical is character-level neural machine translation? assessing mt quality with contrastive translation pairs,” *arXiv preprint arXiv:1612.04629*, 2016.

Appendices

Appendix A

Necessary links mentioned in this report

A.1 List of dataset's data source links

Source Name	Website link
Amazonâs Mechanical Turk	https://www.mturk.com/
Wikipedia	https://www.wikipedia.org/
Khan Academy	https://www.khanacademy.org
Coursera	https://www.coursera.org
Udacity	https://www.udacity.com
TED Talks	http://www.ted.com
ELRA	http://www.elra.org
Banglapedia	https://www.banglapedia.org/
Awake and Watchtower	https://www.jw.org/en/library/magazines/

Table A.1: List of datasets' data source links.

A.2 List of all necessary links mentioned by numbers

Name	Link address
CMSWire [1]	www.cmswire.com/customer-experience/why-machine-translation-matters-in-the-modern-era
Internet in Bangladesh [2]	$en.wikipedia.org/wiki/Internet-in-Bangladesh$
Omniscien [3]	$omniscien.com/?faqs=why-do-i-need-machine-translation$
Bengali Language [4]	$en.wikipedia.org/wiki/Bengali_{l}anguage$
ACM [5]	https://dl.acm.org
IEEExplore [6]	https://ieeexplore.ieee.org/Xplore/home.jsp
Elsevier [7]	https://www.elsevier.com
Springer [8]	https://link.springer.com
Anthology [9]	https://www.aclweb.org/anthology/
Opus [10]	opus.nlpl.eu
Tercom [11]	http://www.cs.umd.edu/s̃nover/tercom/
Pan tree bank [12]	https://www.panl10n.net/
$JW300_2$ [13]	$JW300_2$ has 370,948 English and 2,525,512 Bangla Lines. It isn't properly aligned with sentence by sentence.
Mosesdecoder [14]	https://github.com/moses-smt/mosesdecoder/tree/RELEASE-2.1.1/scripts/tokenizer/tokenizer.perl
ReviewOfCorpora [15]	github.com/masumahmedeesha/reviewOfCorpora

Table A.2: List of all necessary links mentioned by numbers.

Appendix B

Datasets Links

Corpus Name	Dataset link
Opus/global-voices	http://opus.nlpl.eu/GlobalVoices.php
Opus/gnome	http://opus.nlpl.eu/GNOME.php
gnome-org	https://l10n.gnome.org/
WAT/indic-multilingual	http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/index.html
Opus/jw300	http://opus.nlpl.eu/JW300.php
Opus/kde4	http://opus.nlpl.eu/KDE4.php
Opus/open-subtitles	http://opus.nlpl.eu/OpenSubtitles.php
Opus/open-subtitles-alt	http://opus.nlpl.eu/OpenSubtitles-alt-v2018.php
Opus/qed	http://opus.nlpl.eu/QED.php
QCRI	http://alt.qcri.org/resources/qedcorpus/
supara-github	https://github.com/maamumin/SUPara
sipc-github	https://github.com/joshua-decoder/indian-parallel-corpora
Symfony	https://symfony.com/legacy
Opus/tanzil	http://opus.nlpl.eu/Tanzil.php
TanzilNet	http://tanzil.net/trans/
TranslationTanzil	http://tanzil.net/#19:1
Opus/tatoeba	http://opus.nlpl.eu/Tatoeba.php
Opus/ubuntu	http://opus.nlpl.eu/Ubuntu.php
Translation-LaunchPad	https://translations.launchpad.net/

Table B.1: List of dataset links to download available parallel corpora