LANGUAGE TRANSLATION USING NEURAL MACHINE TRANSLATION

NUSRAT JAHAN SUHA

MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

LANGUAGE TRANSLATION BY USING NEURAL MACHINE TRANSLATION

NUSRAT JAHAN SUHA

Thesis submitted in fulfillment of the requirements
For the award of the degree of
Master of Science in Information and Communication Engineering

Department of Information and Communication Engineering

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

January 2021

SUPERVISOR'S DECLARATION

We here by declare that we have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Science in Information and Communication Engineering.

_____

Name of Supervisor: DR. MD. ASHIKUR RAHMAN KHAN

Position: PROFESSOR & CHAIRMAN

Date:

_____

Name of Co-Supervisor: ZAYED-US-SALEHIN

Position: ASSISTANT PROFESSOR

Date:

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations, references and summaries which have been duly acknowledged. This thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

_____

Nusrat Jahan Suha

Roll: BKH1811MSc105F

Date:

Dedicated to my Parents

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Professor Dr. Md. Ashikur Rahman Khan for his germinal ideas, invaluable guidance, continuous encouragement and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science, and his belief that a M.Sc. program is only a start of a life-long learning experience. I appreciate his consistent support from the first day I applied to graduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my native mistakes, and his commitment to my future carrier. I also would like to express very special thanks to my co-supervisor Zayed-Us-Salehin for his suggestions and co-operations throughout the study. I also sincerely thanks for the time spent proofreading and correcting my many mistakes.

My sincere thanks go to all my lab mates, members of the staff of the Information and Communication Engineering Department, NSTU, who helped me in many ways and made my stay at NSTU pleasant and unforgettable.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions which were crucial for the successful completion of this study.

ABSTRACT

Machine translation (MT) is an automatic translation from one language to another. The benefit of machine translation is that it is possible to translate large envelop of text in a very short time. Neural machine translation (NMT) is an approach to machine translation that uses an artificial neural network to predict the possibility of a sequence of words, typically modeling entire sentences in a single integrated model. In this paper, a recurrent neural network named long short term memory is used. Overall neural Machine translation is used for translating not only Bengali languages but also different western and Asian languages to English. Unlike the traditional phrase-based translation system which consists of many small sub-part that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation. In this model, it gives a satisfied BLEU score which quality is better than human translation. Also, a comparison between Google translator and established neural machine translation has happened. These thesis paper have also worked with 4 uncommon languages to translate into English. These languages even do not add to the Google translator. The research has also shown that western languages have given better BLEU score than the Asian language. Especially the Latin script languages have given better translation quality than other script languages. The research has worked with 27 languages. Among 27 languages 14 languages have been widely analyzed. For the poor datasets, the BLEU score is not good but in the larger datasets, it gives satisfactory results than ever. Here, the neural machine translation model is established, trained. And by using these 27 languages datasets the model translation capacity have evaluated. Established neural machine translation gives pretty good translation quality. Although all the languages have not given a higher BLEU score or accuracy like Google translator, the translation quality is good according to the BLEU score matrix algorithm. For some languages the proposed and established model has given a better BLEU score than the Google Translator. Also, some sentences of every language have compared with the output of Google translator. In this comparison, a translation difference between the established translation system and Google translator is found. For some languages, the meaning of the languages is fully reversed in Google translator. But on the other hand, the established neural machine translation system has given a good translation in these aspects. So, in this research different languages are translated to English by the Neural Machine Translation model. This can make a good contribution to the field of machine translation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
RNN	Recurent Neural Network
FNN	Feed Forward Neural Network
NMT	Neural Machine Translation
LSTM	Long Short Term Memory
BLEU	Bilingual Evaluation Understudy
GT	Google Translator
SMT	Statistical machine translation
MT	Machine Translation
EBMT	Example Based Machine Translation
GNMT	Google Neural Machine Translation
IBM	International Business Machines Corporation
AI	Artificial Intelligence
NN	Neural Network
ALPAC	Automatic Language Processing Advisory Committee
RBMT	Rule Based Machine Translation
CPU	Central Processing Unit
GPU	Graphics Processing Unit
OSM	Operation Sequence Model
NNJM	Neural Network Joint Model
NLTK	Natural Language Toolkit
LRC	Linguistics Research Center
HTER	Human-targeted Translation Error Rate

NIST	National Institute of standards and Technology
TER	Translation Error Rate
PER	Position-independent Error Rate
CDER	Cover Disjoint Error Rate
METAL	Mechanical Translation and Analysis of Languages
EM	Expectation-Maximization
OOV	Out-Of-Vocabulary
NECTEC	National Electronics and Computer Technology Center
GRU	Gated Recurrent Unit
IPA	International-Phonetic-Alphabet
UNL	Universal Networking Language
VBMT	Verb-Based Machine Translation
ALPAC	Automatic Language Processsing Advisory Committee

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In this chapter, the introductory study of this thesis is performed. The thesis work is focused on machine translation from Bengali to English and also works on many other languages translated to English. So machine translation, the challenges, the background is focused on this chapter. Hence background study or history on machine translation system is performed first. Then the problem statement is presented and based on this, the research objectives are defined. At the end of this chapter, the outline of this thesis is presented.

1.2 BACKGROUND

Machine translation is the application of computers to the task of translating text from one language to another. However, the translation from one language to another is not an easy task by machine. Many factors contribute to the difficulty of MT including words with multiple meanings, sentences with multiple grammatical structures, and so on. An automatic translator can do the job of translation faster than a human translator that will save a lot of time and cost of money. Here we have used the neural machine translation to translate different languages to English which is used as a recurrent neural network. Neural Machine Translation is a machine translation approach that applies a large artificial neural network toward predicting the possibility of a sequence of words, often in the form of whole sentences. This system is available for the user who is used

in this system for translation. It will make a significant change in the field of the machine translation system. The origin of machine translation first introduced by the work of Al-Kindi, a 9th-century Arabic cryptographer.He developed techniques for systemic language translation, including cryptanalysis, frequency analysis, and probability and statistics, which are used in modern machine translation. In 1629 René Descartes proposed a universal language. In the mid-1930s the first patents for translating machines were applied for by Georges Artsrouni, for an automatic bilingual dictionary using paper tape [1].

In 1933, Soviet scientist Peter Troyanskii presented a paper to the Academy of Sciences of the USSR which is titled as the machine for the selection and printing of words when translating from one language to another. The invention was simple. There were used cards in four different languages, a typewriter, and an old-school film camera. First, the operator took the first word for the text and searched the corresponding card, took a photo.Then, typed its morphological characteristics on the typewriter. The typewriter's keys encoded one of the features. The tape and the camera's film were used simultaneously, making a set of frames with words and their morphology. Despite all this, happened in the USSR, the invention was considered useless. The work of Peter Troyanskii was not known by the world until two Soviet scientists found his patents in 1956. On January 7th 1954, at IBM headquarters in New York started working with machine translation. The IBM 701 computer automatically translated 60 Russian sentences into English for the first time in history. But the translated examples were carefully selected and tested to exclude any ambiguity. For everyday use, that system was no better than a pocket phrasebook. Canada, Germany, France, and especially Japan, all joined the race for machine translation. The vain struggles to improve machine translation lasted for forty years. In 1966, the US ALPAC committee, in its famous report, called machine translation expensive, inaccurate, and unpromising. They instead recommended focusing on dictionary development, which eliminated US researchers from the race for almost a decade. Not all countries had the same views as ALPAC. In the 1970s, Canada developed the METEO system, which translated weather reports from English into French. It was a simple program that was able to translate 80,000 words per day. The program was successful enough to enjoy use in the 2000s before requiring a system update. The French Textile Institute used machine translation to convert abstracts from French into English, German, and

Spanish. Around the same timeframe, Xerox used their system to translate technical manuals. Both were used effectively as early as the 1970s, but machine translation was still only scratching the surface by translating technical documents. By the 1980s, people were diving into developing translation memory technology, which was the beginning of overcoming the challenges posed by nuanced verbal communication. But, systems continued to face the same trappings when trying to convert text into a new language without losing meaning. Due to the creation of the Internet and all the opportunity it offered, Franz-Josef Och won a machine translation speed competition in 2003 and would become head of Translation Development at Google. By 2012, Google announced that its own Google Translate translated enough text to fill one million books in a day. Japan also leads the revolution of machine translation by creating speech-to-speech translations for mobile phones that function for English, Japanese, and Chinese. This is a result of investing time and money into developing computer systems that model a neural network instead of memory-based functions. As such, Google informed the public in 2016 that the implementation of a neural network approach improved clarity across Google Translate, eliminating much of its clumsiness. There discussed different machine translation system and how it works today. The first ideas surrounding rule-based machine translation appeared in the 70s.The scientists peered over the interpreters' work, trying to compel the tremendously sluggish computers to repeat those actions. These systems consisted of Bilingual dictionary and a set of linguistic rules for each language, PROMPT and Systran are the most famous examples of RBMT systems. But even they had some nuances and subspecies. This is the most straightforward type of machine translation is named direct machine translation. It divides the text into words, translates them, slightly corrects the morphology, and harmonizes syntax to make the whole thing sound right, more or less. The output returns some kind of translation. Usually, it's quite crappy. It seems that the linguists wasted their time for nothing. Modern systems do not use this approach at all, and modern linguists are grateful. In contrast to direct translation, transfer-based machine translation prepare first by determining the grammatical structure of the sentence. Then it manipulates whole constructions, not words, afterwards. This helps to get a quite decent conversion of the word order in translation. In practice, it still resulted in literal translation and exhausted linguists. On the one hand, it brought simplified general grammar rules. But on the other, it became more complicated because of the increased

number of word constructions in comparison with single words.In Interlingua based machine translation; the source text is transformed into the intermediate representation and is unified for all the world's languages. It's the same Interlingua Descartes dreamed of a Meta language, which follows the universal rules and transforms the translation into a simple back and forth task. Next, Interlingua would convert to any target language, and here was the singularity! Because of the conversion, Interlingua is often confused with transfer-based systems. The difference is the linguistic rules specific to every single language and Interlingua, and not the language pairs. This means it is possible to add a third language to the Interlingua system and translate between all three. This cannot be done in transfer-based systems. It looks perfect, but in real life, it's not. It was extremely hard to create such universal Interlingua. A lot of scientists have worked on it their whole lives. They've not succeeded. Japan was especially interested in fighting for machine translation. There were reasons, very few people in the country knew English. It promised to be quite an issue at the upcoming globalization party. So the Japanese were extremely motivated to find a working method of machine translation. Rule-based English-Japanese translation is extremely complicated. The language structure is completely different, and almost all words have to be rearranged and new ones added. In 1984, Makoto Nagao from Kyoto University came up with the idea of using ready-made phrases instead of repeated translation. The more examples we have, the better the translation. EBMT showed the light of day to scientists from all over the world: it turns out, you can just feed the machine with existing translations and not spend years forming rules and exceptions. In early 1990, at the IBM Research Center, a machine translation system was first shown which knew nothing about rules and linguistics as a whole, which is called statistical machine translation. It analyzed similar texts in two languages and tried to understand the patterns. The method was much more efficient and accurate than all the previous ones. And no linguists were needed. The more texts we used the better translation we got. The machine needed millions and millions of sentences in two languages to collect the relevant statistics for each word. In the beginning, the first statistical translation systems worked by splitting the sentence into words, since this approach was straightforward and logical. IBM's first statistical translation model was called Model One. Model one used a classical approach to split into words and count stats. The word order wasn't taken into account. The only trick was translating one word into multiple words. Model 2 considers the word order in

sentences. The lack of knowledge about languages word order became a problem for Model 1, and it's very important in some cases. Model 2 dealt with that. It memorized the usual place the word takes at the output sentence and shuffled the words for the more natural sound at the intermediate step. Things got better, but they were still kind of crappy. Model 3 considers the necessity of a new word choosing the right grammatical particle or word for each token-word alignment.Model 2 considered the word alignment, but knew nothing about the reordering. For example, adjectives would often switch places with the noun and no matter how good the order was memorized; it wouldn't make the output better. Therefore, Model 4 took into account the so-called relative order the model learned if two words always switched places. Model 5 fixes bug, nothing new here. Model 5 got some more parameters for the learning and fixed the issue with conflicting word positions. Despite their revolutionary nature, word-based systems still failed to deal with cases, gender, and homonymy. Every single word was translated in a single-true way, according to the machine. Such systems are not used anymore, as they've been replaced by the more advanced phrase-based methods. This method is based on all the word-based translation principles through statistics, reordering, and lexical hacks. Although for the learning, it split the text not only into words but also phrases. These were the n-grams, to be precise, which were a contiguous sequence of n words in a row. Thus, the machine learned to translate steady combinations of words, which noticeably improved accuracy. Besides improving accuracy, the phrase-based translation provided more options in choosing the bilingual texts for learning. For the word-based translation, the exact match of the sources was critical, which excluded any literary or free translation. The phrase-based translation had no problem learning from them. To improve the translation, researchers even started to parse the news websites in different languages for that purpose. This method should also be mentioned, briefly. Many years before the emergence of neural networks, syntax-based translation was considered the future of translation but the idea did not take off. The proponents of syntax-based translation believed it was possible to merge it with the rule-based method. It's necessary to do quite a precise syntax analysis of the sentence to determine the subject, the predicate and other parts of the sentence, and then to build a sentence tree. Using it, the machine learns to convert syntactic units between languages and translates the rest by words or phrases. That would have solved the word alignment issue once and for all. The problem is the syntactic parsing works terribly even though

we consider it solved a while ago.The regular RNN was used at the beginning, then upgraded to bi-directional where the translator considered not only words before the source word, but also the next word. That was much more effective. Then it followed with the hardcore multilayer RNN with LSTM-units for long-term storing of the translation context. In two years, neural networks surpassed everything that had appeared in the past 20 years of translation. Neural translation contains 50% fewer word order mistakes, 17% fewer lexical mistakes, and 19% fewer grammar mistakes [1]. The neural networks even learned to harmonize gender and case in different languages. And no one taught them to do so. The most noticeable improvements occurred in fields where direct translation was never used. Statistical machine translation methods always worked using English as the key source. Thus, if there needs any translation from Russian to German, the machine first translate the text to English and then from English to German, which leads to a double loss. Neural translation doesn't need that — only a decoder is required so it can work. That was the first time that direct translation between languages with no common dictionary became possible. So, there still a long way to go in the field of machine translation.

1.3 PROBLEM STATEMENT

Machine translation is the task of automatically converting source text in one language to text in another language. In a machine translation task, the input already consists of a sequence of symbols in some language and the computer program must convert this into a sequence of symbols in another language. Given a sequence of text in a source language, there is no single best translation of that text to another language. This is because of the natural obscurity and flexibility of human language. This makes the challenge of automatic machine translation difficult, perhaps one of the most difficult in artificial intelligence. Some of the important problems are discussed below that are associated with machine translation.

Quality Issues: Quality issues are perhaps the biggest problems when using machine translation. No computer software can process the context in which a language is being used and thus to which the translation needs to occur. With tens of thousands of words in every language and several different meanings to thousands of words based on the

context of a word being used, a computer program can't understand the circumstance, especially in complex situations. Humans, on the other hand, can understand the language in the circumstance it is used. This is because the human can understand emotions, silent communication, and culture, which all affect the context of language as it is used. Without being able to understand all of these factors, machine translation will always have problems with the quality of its translation.

Lack of Creativity: It is said that language is an art that can only be mastered after thousands of hours of exposure and practice. Though machine translation does a decent job of translating some subject matter, it most surely can't master the art of language which involves a lot of creativity. This creativity is important to understand when communicating with business partners or clients within the global market and also when communicating the message to potential clients in that same market. Human translators with expertise and experience in both languages at hand will be able to be more creative with the subject matter at hand and thus deliver a more creative solution that will encourage business partners and customers.

Lack of Sensitivity: Each country has its own culture which in turn has their values and standard that is followed. The values and standard of each culture influence the way people within that culture communicate with each other. Without having experience with the culture, it is impossible to know which values and norms to be sensitive to when communicating. Machine translation cannot be sensitive to cultures values or standard. This can cause problems by delivering the wrong message to business associates or customer, or even worse offending them when we're not meaning to. This could potentially cause many problems which can be very costly as it will affect performance in the foreign market.

Multiple Meanings in Translation: Often a word has a different meaning. It cannot be done by the machine to detect all the meaning always. So, then people cannot get the proper translation. The same word may mean multiple things depending on where it's placed and how it's used in a sentence.

Translating Language Structure: Every language uses a defined structure with own agreed-upon rules. The complexity and singularity of this framework directly correlate to the difficulty of translation. A simple sentence in English has a subject, verb, and object — in that order. For example, "she eats rice." But not every language shares this structure. Farsi typically follows a sequence of a subject, then object, then verb. And in

Arabic, subject pronouns become part of the verb itself. As a result, translators frequently have to add, remove, and rearrange source words to effectively communicate in the target language.

Translating Idioms and Expressions: Idiomatic expressions explain something by way of unique examples or figures of speech. And most importantly, the meaning of particular phrases cannot be predicted by the accurate definitions of the words it contains. Many linguistic professionals insist that idioms are the most difficult items to translate. Idioms are routinely mentioned as a problem machine translation engines will never fully solve.

Missing Names in Translation: A language may not have an exact match for a certain action or object that exists in another language. In American English, for instance, some homeowners have what they describe as a "guest room." It is simply a space where their guests can sleep for the night. This concept is common in other languages as well but often expressed quite differently. Greeks describe it with the single word "ksnona" while their Italian neighbours employ a three-word phrase "camera per gliospiti" instead. This is the first step towards localization. Also if we think about greeting then it varies from languages to languages.

Translating Sarcasm: Sarcasm is a sharp style of expression.It is usually means the opposite of its line-for-line phrasing. Sarcasm frequently loses its meaning when translated word-for-word into another language and can often cause inappropriate misunderstandings. Preferably, a publisher would remove sarcasm from the source text before translation. But in cases where that sarcasm is central to the content requirements, the publisher should clearly emphasize sarcastic passages. That way, translators will have a chance to avoid word for word misunderstandings and suggest a local idiom that may work better in the target language.

NMT sucks with small datasets: While this can be said about most machines learning in general, this problem is especially pronounced in NMT. The promise of NMT is it generalizes better (than phrase-based MT) with increasing data. But under low data situations, NMT does significantly worse. For poor datasets, it gives an unrelated output that has no connection with the output.

NMT performs badly with out-of-domain data: Current machine translation systems produce very fluent outputs unrelated to the input for out-of-domain data. So a general MT system like Google Translate will do particularly badly in specialized domains like

legal or finance. Other traditional machine translations do better than the neural machine translation system in this aspect.

NMT performs poor for rare words: While still better than phrase-based translation, NMT does poorly for rare words. This can become a problem for highly-Phonetics languages and domains with a lot of named entities where the Phonetics forms and named entity occurrences respectively can be very rare. Encoding long sentences and generating long sentences is still a vast problem. MT systems decay with sentence length, but NMT systems especially so. Using attention helps but the problem is far from "solved". In many domains, such as legal, it is quite common to have long, complex sentences. But for the long or complex sentences, the result is not satisfactory.
There are a lot of reasons that machine translation cannot be replaced by human translation. Some of them are given below.

i. Not all the words in one language have equivalent words in another language. In some cases, a word in one language is to be expressed by the group of words in another.
ii. Two given languages may have completely different structures. Different languages have different structure.
iii. Sometimes there is a lack of one-to-one correspondence of parts of speech between two languages. For example, the colour terms of Tamil are nouns whereas in English they are adjectives.
iv. The ways sentences are put together also differ among languages.
v. Words can have more than one meaning and sometimes a group of words or whole sentence may have more than one meaning in a language. This problem is called ambiguity.
vi. Not all translation problems can be solved by applying values of grammar.
vii. It is too difficult for software programs to predict meaning.
viii. Translation requires not only vocabulary and grammar but also knowledge gathered from experience.
ix. The programmer should understand the rules under which complex human language operates and how the mechanism of this operation can be simulated by automatic means.

x. The simulation of human language behaviour by automatic means is almost impossible to achieve as the language is an open and dynamic system in constant change. More importantly, the system is not yet completely understood.

1.4 OBJECTIVES

There are some objectives along with the research. Although language translation or converting text from one language to another is a challenging and complicated task, with study and motivation it can be more available for the common people. And people will get the benefit of a machine translation system.

i. To study the machine translation system,
ii. To improve the accuracy of the Machine translation system
iii. To use Machine Translation system in Education
iv. To work with the rare languages in machine translation.
v. To establish a model that is good for translation from different languages to English.
vi. To Work with problems associated with the machine translation system.

1.5 SCOPE OF THE STUDY

Neural Machine Translation is a machine translation approach that applies a large artificial neural network toward predicting the likelihood of a sequence of words, often in the form of whole sentences. Recurrent Neural Networks are different from traditional ANNs because they don't need a fixed input/output size and they also use previous data to make predictions. Here used Neural Machine translation where used LSTM. Translation of different languages to English is the main scope of this study. Many methods are used for machine translation from different languages to English but neural machine translation is the least used. So, in this thesis, there used NMT to translate different languages (Bengali, Marathi, Thai, Arabic, Romanian) to English. Along their also discussed and analyzed the problems of the machine translation system and discussed how this can be overcome. As discussed before machine translation has a lot of opportunities to work. This is a potential field with many more problems. By

solving this problem these can do better in the field of machine translation. Here, not only from Bengali to English but also from translating other languages to English has also shown. The result of the established model also compares the translation and Google translate. The required tools carry out to work this work is:

- Google Co-Lab
- Python 3.0
- Python Packages(NLTK, Keras, Tensor Flow, pickle, NumPy)
- Strong CPU needed and further processing need GPU

1.6 OVERVIEW OF THE THESIS

Chapter 1 - Introduction: Performs the background of the Machine Translation system. In the addition problem statement, the scope of the study is presented.

Chapter 2 - Literature Review: An illustrative review of related work is performed on machine translation is discussed.

Chapter 3 – Methodology: Presents approaches that are used in the system. And also the dataset details what function and algorithm are used are discussed.

Chapter 4 - Implementation: Show process of the whole translation process, text cleaning, train model, test model and also evaluate the model and its translation quality. All things are established by Google Co-Lab and Microsoft Excel. Python Language is used.

Chapter 5 - Results and Discussion: Here Discussed the results and analyzed it how it affects the machine translation research. Showed the performance of the established Machine Translation model and also compared it with the Google translator.

Chapter 6 - Conclusions: This chapter describes the conclusion and future research that will happen according to this model. Also, the shortcoming of this research has also told here.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, there have shown the previous work that has been done according to or similar to this research work. Also, there has discussed other types of methods that are used in MT. There is a lot of work that has been done in MT, also happening right now. For Google translate accuracy is another concern of consideration. Different researchers have worked on different types of work, their advantages, disadvantages also showed here in a short

.2.2 DIFFERENT TYPES OF MACHINE TRANSLATION SYSTEM

Galley, M. and Manning, C.D show that A novel hierarchical phrase reordering model can successfully handle the key examples often used to motivate syntax-based systems, such as the rotation of a prepositional phrase around a noun phrase[2]. In [3], statistical machine translation system performed successfully in previous DARPA competitions on open-domain text translations. The author achieved the highest BLEU score in 2 out of 5 languages pairs and had competitive results for the other language pairs. Another paper [4] proposes a statistical machine translation model that uses hierarchical phrases—phrases that contain sub phrases. The model is formally a synchronous context-free grammar but is learned from a parallel text without any syntactic annotations. Thus it can be seen as combining fundamental ideas from both syntax-based translation and phrase-based translation. The system trains and decodes methods in detail and evaluates them for translation speed and translation accuracy.

Using BLEU as a metric of translation accuracy, a state-of-the-art phrase-based system is found. In [5] proposes to extend basic encoder decoder architecture by allowing a model to automatically search for parts of a source sentence. The source sentences are relevant to predicting target words. Popovic, M., Ney, H[6] investigate new possibilities for improving the quality of statistical machine translation by applying word reordering of the source language sentences based on Part-of-Speech tags. Results are presented on the European Parliament corpus containing about 700k sentences and 15M running words. To investigate sparse training data scenarios, results obtained on about 1% of the original corpus. The source languages are Spanish and English and the target languages are Spanish, English and German. Two types of reordering depending on the language pair and the translation direction is proposed: local reordering of nouns and adjectives for translation from and into Spanish and long-range reordering of verbs for translation into German. For the best translation system achieve up to 2% relative reduction of WER and up to 7% relative increase of BLEU score. In the paper [7] addresses the problem of reordering in statistical machine translation. An elegant and efficient approach to couple reordering and decoding described, which does not need any additional model. A novel machine translation model, the Operation Sequence Model combines the benefits of phrase-based and N-gram-based SMT and remedies their drawbacks are described in [8]. The model represents the translation process as a linear sequence of operations that includes not only translation operations but also reordering operations.Their method [9] uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. The author's main result is that on an English to french translation task from the the WMT-14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. The described approach [10] achieves significant word error reductions concerning a carefully tuned 4-gram back off language model in a state of the art conversational speech recognizer for the DARPA rich transcriptions evaluations. A new algorithm for efficient training an unsmoothed error count is presented in [11]. They show that significantly better results can often be obtained if the final evaluation criterion is taken directly into account as part of the training procedure. In [12] there uses off-the-shelf linear binary classifier software and can be built on top of an existing MERT framework in a matter of

hours.They establish PRO's scalability and effectiveness by comparing it to MERT and MIRA. It demonstrates parity on both phrase-based and syntax-based systems in a variety of language pairs, using large scale data scenarios. Ondrej Bojar et al.[13] presents the results of the WMT16 shared tasks.The task is included five machine translation tasks, three evaluation tasks, an automatic post-editing task and bilingual document alignment task. Milos Stanojevi ˇc´, Khalil Sima'an[14] evaluate their metric on the standard WMT12 data showing that it outperforms the strong baseline METEOR. They also analyze the contribution of individual features and the choice of training data, language-pair vs. target-language data, providing new insights into this task. In [15] a novel semi-automated metric, MEANT assesses translation utility by matching semantic role fillers, producing scores that correlate with human judgment as well as HTER but at a much lower labour cost. The results show that their proposed metric is significantly better correlated with human judgment on adequacy than current widespread automatic evaluation metrics while being much more cost-effective than HTER. Bojar, O et al.[16] presents the results of the WMT16 Metrics Shared Task. They asked participants of this task to score the outputs of the MT systems involved in the WMT16 Shared Translation Task. They collected scores of 16 metrics from 9 research groups. In addition to that, they computed scores of 9 standard metrics (BLEU, Sent BLEU, NIST, WER, PER, TER and CDER) as baselines. In the thesis paper [17] investigate the use of monolingual data for NMT. The work explores strategies to train with monolingual data without changing the neural network architecture.Here, [18] present a novel formulation for a neural network joint model which augments the NNLM with a source context window. The model is purely lexicalized and can be integrated into any MT decoder. The authors also present several variations of the NNJM which provide significant additive improvement. In [19] paper, offer an in-depth analysis of the modelling and search performance. The reasearch addresses the question of a more complex search algorithm is necessary. Furthermore, complex models are investigated which might only be applicable during rescoring are promising. These results indicate that the current search algorithms are sufficient for NMT systems. The works [20] introduce a neural machine translation (NMT) model that maps a source character sequence to a target character sequence without any segmentation.In this multilingual setting, the character-level encoder significantly outperforms the sub-word-level encoder on all the language pairs. The authors observe that on CS-EN, FI-EN and RU-EN surpasses the models

specifically trained on that language pair alone. The results are found in terms of BLEU score and human judgment. In [21] paper, introduces a simpler and more effective approach. The approach makes the NMT model capable of open-vocabulary translation by encoding rare and unknown words as sequences of subword units. Subword models improve over a back-off dictionary baseline for the WMT 15 translation tasks English-German and English-Russian by up to 1.1 and 1.3 BLEU.The paper [22] present and compare various methods for computing word alignments using statistical or heuristic models. Five alignment models presented in Brown et al (1993), the hidden Markov alignment model, smoothing techniques, and refinements are considered. In the paper [23] Apertium a free, open-source platform for rule-based machine translation is discussed.It is being widely used to build machine translation systems for a variety of language pairs,where shallow transfer suffices to produce good quality translations. It has also proven useful in assimilation scenarios with more distant pairs involved. In [24] present a conversion system for Punjabi text to its equivalent Punjabi dialect text which could be extended easily to be applied to other dialects of Punjabi. The proposed system is developed using the rule-based approach that mainly relies on morphological analysis, conversion rules and bilingual dictionaries.METAL, a fully automatic high-quality Machine Translation (MT) system is discussed [25]. After outlining the history and status of the project, this paper discusses the system's projected application environment and briefly describes its general translation approach. After detailing the salient linguistic and computational techniques on which METAL is based, some of the practical aspects of such an application are considered. Experimental results that imply the system is now ready for production use are included also. Two exhibits are appended: a German original text and its raw METAL translation.In this paper, [26] describes the components of statistical machine translation system. This system combines phrase-to phrase translations extracted from a bilingual corpus using different alignment approaches. Special methods to extract and align named entities are used. The authors show how a manual lexicon can be incorporated into the statistical system in an optimized way. Experiments on Chinese-to-English and Arabic-to-English translation tasks are presented. A comprehensive conversion system to convert the Chittagonian dialect to the standard Bengali language is presented [27]. The system is a text to text conversion system based on word-to-word mapping. The conversion system adopts a bilingual dictionary, rule-based morphological transformation on suffixes, and a

supportive word suggestion module. The system tokenizes the regional input text and processes the tokens through word-to-word mapping and morphological transformation using suffix transformation rules if word-to-word mapping fails. The authors have also introduced an aiding tool that generates suggested words for the dialectal input. The system achieved an accuracy of 94.75% for producing standard Bengali translation from Chittagonian words. In [28], the paper compares the 3 methods of estimating word translation probabilities for selecting the translation word in Thai – English Machine Translation. The 3 methods are (1) Method based on the frequency of word translation, (2) Method based on collocation of word translation and (3) Method based on Expectation-Maximization (EM) algorithm. For evaluation, Thai – English parallel sentences generated by NECTEC is used. The method based on the EM algorithm is the best in comparison to the other methods and gives satisfying results.

2.3 BENGALI MACHINE TRANSLATION

A technique to translate Bengali sentence into corresponding English sentence using context-sensitive grammar rules is proposed [29]. A set of context-sensitive grammar rules is proposed to translate Bengali sentences including assertive, interrogative and imperative sentences.The evaluation result reveals that the proposed machine translation system can translate Bengali sentences into English with 83% accuracy and gives better accuracy compared to the Google translator for some selected sentences. The work [30] represents a new solution for building an MT system for English to Bengali translation. The rule-based transfer approach of the MT system is modified for this solution. In machine translation the searching of word from the lexicon is a compulsory task. Here this searching stage is utilized efficiently by proposing an intelligent integer-based lexicon system. The system consists of several separate lexicons and an algorithm is also developed for searching words from the lexicon to accomplish the basic steps of machine translation. In [31] a model of transfer architecture has been proposed which represents a Rule-Based Approach. This approach relies on fuzzy rules. It is tense and phrase-based English to the Bengali transfer system. Comparing the experimental result with Google translator, it has been found that the model translation system provides higher accuracy than comparing translator. This [32] paper presents the adapting rule-based machine translation from English to Bengali. The proposed language translation

model relies on rule-based methodologies, especially fuzzy rules. There are “If-Then” basis rules are applied for the English to Bengali language translation. In this language translation, we use the rough set technique for Knowledge Representation System. This technique is used to classify each English sentence to a particular class using attributes of that English sentence and then translate them to the Bengali sentence using the rules that are produced earlier in the language translation system. Here English to Bengali bilingual dictionary has been formed for language translation. In[33] the proposed system the source text is analyzed using a set of grammatical rules, transferred and synthesized with the direct help of some dictionaries (that is lexicons).In this system, a Word Corresponding Lexicon in addition to the Root Lexicon and the Suffix Lexicon and a typical set of Bengali to English transfer rules are used. The work [34] has discussed the development procedure of machine translation (MT) dictionaries. This includes determining the contents of the dictionaries, determining the detail of information, and determining the organization of the dictionaries. Some of the information about Bengali words that should be included in an MT dictionary for Bengali has been presented. Some Bengali morphological rules have also presented. This paper [35] describes the architecture of the English to Bengali machine translation system. The system has been implemented with the encoder-decoder recurrent neural network. The model uses a knowledge-based context vector for the mapping of English and Bengali words. From the execution of the GRU and LSTM layer, GRU performed better than LSTM. In this work [36], the authors describe a phrase-based Statistical Machine Translation (SMT) system that translates English sentences to Bengali. A transliteration module is added to handle out-of-vocabulary (OOV) words.The improvement of our system through effective impacts on the BLEU, NIST and TER scores have shown also. The overall BLEU score of the system is 11.7 and for short sentences, it is 23.Due to the small available English- Bengali parallel corpus, the Example-Based Machine Translation (EBMT) system has a high probability of handling unknown words. To improve translation quality for the Bengali language, the paper [37] proposes a novel approach for EBMT using WordNet and International-Phonetic-Alphabet (IPA)-based transliteration. The proposed system first tries to find semantically related English words from WordNet for the unknown word. The work [38] focuses on morphological analysis of Bengali words to incorporate them into Bengali to Universal Networking Language (UNL) processors. In [39], the work

particularly emphasizes bringing previous works on morphological analysis in the framework of UNL, to produce a Bengali -UNL dictionary. A method to analyze syntactically Bengali sentence using context-sensitive grammar rules which accept almost all types of Bengali sentences including simple, complex and compound sentences. Then it interprets the input Bengali sentence to English using the NLP conversion unit. The effectiveness of this method has been justified over the demonstration of different Bengali sentences with 28 decomposition rules and the success rates in all cases are over 90%. The work [40] presents a tense based Machine Translation (MT) system which operates on English sentences as an input language. The MT system translates the input into its corresponding Bengali sentences using the Natural Language Processing (NLP) conversion technique. The system uses context-free grammars for validating the syntactical structure of the input sentences and for parsing it applies the bottom-up approach. To verify the accuracy of the generated output sentences is happened by this system.In [41] proposes verb-based machine translation (VBMT), a new approach to machine translation (MT) from English to Bengali (EtoB). For translation, the system simplifies any form (simple, complex, compound, active and passive form) of an English sentence into the simplest form of an English sentence that is subject plus verb plus object. Maxim Roy, Fred popwich [42] apply several word reordering techniques to a Bengali -English Statistical Machine Translation (SMT) system. The authors evaluate the approaches through their impact on the BLEU score for the phrase-based Bengali -English SMT system .The system also provides a new test set with multiple references between Bengali and English for SMT evaluation purposes and evaluate the reordering approaches on the extended test set.

2.4 NEURAL MACHINE TRANSLATION SYSTEM

In [43], the potential and the limitation of current machine translation are discussed by comparing the output of human translation and that of virtual machine translation. Here, “virtual machine translation” means a kind of syntax-oriented literal translation which may be regarded as an idealized competence of today's practical machine translation. The paper [44] presents the experimental results of several approaches to machine translation evaluation to determine the quality of Thai to English translation. In [45] the authors focus on the current scenario of research in machine translation in

India. They have reviewed various important Machine Translation Systems (MTS) and presented a preliminary comparison of the core methodology as used by them. The literature shows that there have been many attempts in MT for English to Indian languages and Indian languages to Indian languages. At present, a number of government and private sector projects are working towards developing a full-fledged MT for Indian languages.Wu, Yonghui, et al. [46] gives a brief description of the various approaches and major machine translation developments in India .In this work, they present GNMT, Google's Neural Machine Translation system, which attempts to address many of these issues. A model consists of a deep LSTM network with 8 encoder and 8 decoder layers are used. The model has used residual connections as well as attention connections from the decoder network to the encoder. In particular, there has been no comprehensive analysis of how well Google Translate (GT) performs, perhaps the most used system. In [48] investigate the translation accuracy of 2,550 language-pair combinations provided by this software. Results show that the majority of these combinations provide adequate comprehension, but translations among Western languages are generally best and those among Asian languages are often poor. Another thesis paper of the same author [49] shows that translations between English and German, Afrikaans, Portuguese, Spanish, Danish, Greek, Polish, Hungarian, Finnish, and Chinese tend to be the most accurate. The work [50] shows that an improved Bleu score is neither necessary nor sufficient for achieving an actual improvement in translation quality, and give two significant counterexamples to Bleu's correlation with human judgments of quality. This offers new potential for research which was previously deemed unpromising by an inability to improve upon Bleu scores. Firat et al's [51] work proposes multi way, multilingual neural machine translation. The proposed approach enables a single neural translation model to translate between multiple languages, with a number of parameters that grows only linearly with the number of languages. The thesis paper [52] proposes a simple solution to use a single Neural Machine Translation (NMT) model to translate between multiple languages. The solution requires no changes to the model architecture from a standard NMT system but instead introduces an artificial token at the beginning of the input sentence to specify the required target language. Using a shared work piece vocabulary enables Multilingual NMT systems using a single model. The models can also learn to perform implicit bridging between language pairs never seen explicitly during training, showing that

transfer learning and zero-shot the translation is possible for neural translation. There showed analyses that hint at a universal Interlingua representation in their models and also show some interesting examples when mixing languages. In the paper [53], investigate the problem of learning a machine translation model that can simultaneously translate sentences from one source language to multiple target languages. The solution is inspired by the recently proposed neural machine translation model which generalizes machine translation as a sequence learning problem. A method of automatic machine translation evaluation that is quick, inexpensive, and language-independent is proposed here[54]. The method correlates highly with human evaluation and that has a little marginal cost per run. This method is presented as an automated understudy to skilled human judges when there is a need for quick or frequent evaluations. In the paper [55], explore different NMT algorithms - Bidirectional Long Short Term Memory (LSTM) and Transformer based NMT, to translate the Bangla to English language pair. For the experiments, different datasets and their experimental results is used to outperform the existing performance. The factors affecting the data quality and how they influence the performance of the models is also investigated. The research paper [56], study the problem of employing a neural machine translation model to translate Arabic dialects to Modern Standard Arabic. The proposed solution of the neural machine translation model is prompted by the recurrent neural network-based encoder-decoder neural machine translation model that has been proposed recently. The work proposes the development of multitask learning (MTL) model which shares one decoder among language pairs, and every source language has a separate encoder. In the thesis paper [57] the translation results within dataset show that the neural-network-based translation got the best BLEU score in average for 0.43. The result analysis indicates that the neural-network-based translation can translate better for Thai sentences containing unknown words and those with numerical classifier expression. An NMT system usually has to apply a vocabulary of certain size to avoid the time-consuming training and decoding, thus it causes a serious out-of-vocabulary problem. Furthermore, the decoder lacks a mechanism to guarantee all the source words to be translated and usually favors short translations, resulting in fluent but inadequate translations. In order to solve the above problems, in the paper [58] incorporate statistical machine translation (SMT) features, such as a translation model and an n-gram language model, with the NMT model under the log-linear framework.The proposed method shows the

significantly improves the translation quality of the state-of the-art NMT system on Chinese-to-English translation tasks. The method produces a gain of up to 2.33 BLEU score on NIST open test sets.Below there has shown some papers with overall concern of the study that is happened in this research.

Author	Title	Method/Process	Limitations
Chiang, D(2007)	Hierarchical phrase-based translation	Present a statistical machine translation model that uses hierarchical phrases—phrases that contain sub phrases.	Only scratched the surface of the modeling challenge
Franz Josef Och (2003)	Minimum error rate training in statistical machine translation	Describe a new algorithm for efficient training an unsmoothed error count	Only work with the Chinese-English dataset.
Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. and Makhoul(2014)	Fast and robust neural network joint models for statistical machine translation	Present a novel formulation for a neural network joint model(NNJM)	There are large areas of research yet to be explored
Nusai, C., Suzuki, Y. and Yamazaki, H.,(2008)	Estimating word translation probabilities for Thai–English machine translation using EM Algorithm	Frequency of word translation, Collocation of word translation, and Expectation-Maximization (EM) algorithm. .	The current experimental setup is restricted to nouns
Mohammed Safayet Arefin, Mohammed Moshiul Hoque, Md. Obaidur Rahman, Md. Shamsul Arefin(2015)	A machine translation framework for translating Bengali assertive, interrogative and imperative sentences into English	Translate Bengali sentence into corresponding English sentence using Context-sensitive grammar rules	For long sentences, the success rate is not satisfactory.

Md. Golam Rabiul Alam,Md Monirul Islam,Nowrin Islam(2011)	A New Approach To Develop An English To Bangla Machine Translation System	Represents a new solution for building an MT system for English to Bengali translation, by modifying the rule-based system	The proposed system is not capable of translating all type of English sentence.
Judith Francisca,Md. Mamun Mia, Dr. S. M. Monzurur Rahman(2011)	Adapting rule based machine translation from English to Bengali	Shown a new approach for language translation.	Have not worked on all the variety of sentences.
Md. Shahnur Azad Chowdhury(2013)	Developing a Bengali to English Machine Translation System Using Parts Of Speech Tagging: A Review	Reviews an efficient implementation of Machine Translation (MT) System from Bengali to English.	This is an efficient system for simple sentences.
Shaykh Siddique, Tahmid Ahmed, Md. Rifayet Azam Talukder, and Md. Mohsin Uddin(2020)	English to Bengali Machine Translation Using Recurrent Neural Network	The system has been implemented with the encoder-decoder recurrent neural network. .	Performance varies dealing with a large amount of vocabulary. .
Kanija Muntarina,Md. Golam Moazzam and Md. Al-Amin Bhuiyan(2013)	Tense based English to Bengali translation using MT system	Presents a tense based Machine Translation (MT) system which operates on English sentences as an input language.	Some indefinite complex structure rules have been ignored here.
Firat, O., Cho, K., Sankaran, B., Vural, F.T.Y. and Bengio, Y.,(2017)	Multi way, multilingual neural machine translation	Proposed multi-way, multilingual neural machine translation. .	Avoiding independent attention mechanism.
Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G. and Hughes, M.,(2017)	Google's multilingual neural machine translation system: Enabling zero-shot translation.	A single Neural Machine Translation (NMT) model to translate between multiple languages.	Slightly lower translation quality

Dong, D., Wu, H., He, W., Yu, D. and Wang, H.,(2015)	Multitasking learning for multiple language translation	A unified neural machine translation model under multitask learning framework.	A general framework for translating from one source language to many targets. .

2.5 SUMMARY

In this chapter, different types of research work have discussed here. There has given a brief discussion about the research study of machine translation. The different machine translation system is used at different times. Along with the need for accuracy and evaluation, machine translation has stepped now in neural machine translation. In the first part, different types of a translation system that are applied in different thesis paper are discussed. Where the problems associated with every system has discussed. For translating Bengali to English different machine translation methods are used but the neural machine translation methods are not used in much. But NMT is the most reliable compared with another translation system. The most popular translator that is used today is Google translator. Google translator used neural machine translation which has now given better result than before. The most popular matrix algorithm BLEU for evaluating translation is used in this research. So, this BLEU algorithm is discussed also according to the thesis paper. Google translation accuracy is another concern of our research. Although Google translator has given better result than before and the accuracy is increasing but it is not fully satisfying for all types of language translation. The accuracy of the google translator is discussed based on a few thesis papers.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In these chapter discussed the methodology that is used for this research. Here, the neural translation model is used for the translation process. In the Neural translation model, the most popular recurrent neural network long short term memory is used. Different types of activation function and algorithm is used. As an activation function Softmax is used. In the translation process, for the optimization algorithm, ADAM has been used. ADAM has given the best result. For evaluation, the most popular matrix algorithm BLEU is used here. All methods that are utilized in this research have been discussed**.**

3.2 WORKFLOW DIAGRAM

First, the workflow diagram of the overall process has been shown here. Here the overall process has shown in one glimpse. The dataset has to be prepared for using it in the model. Raw dataset made in a way that it can be used for the translation and then we have to split the dataset for training and testing. Raw data is a set of information that was delivered from a certain data entity to the data provider and hasn't been processed yet by machine or human. In a dataset, a training set is implemented to build up a model, while a test set is to validate the model built. So, we use the training data to fit the model and testing data to test it. We have established the NMT model for the research. We have trained the model with the proper steps. our model have measured translation quality by using the BLEU score. These are the overall workflow diagram of the process.

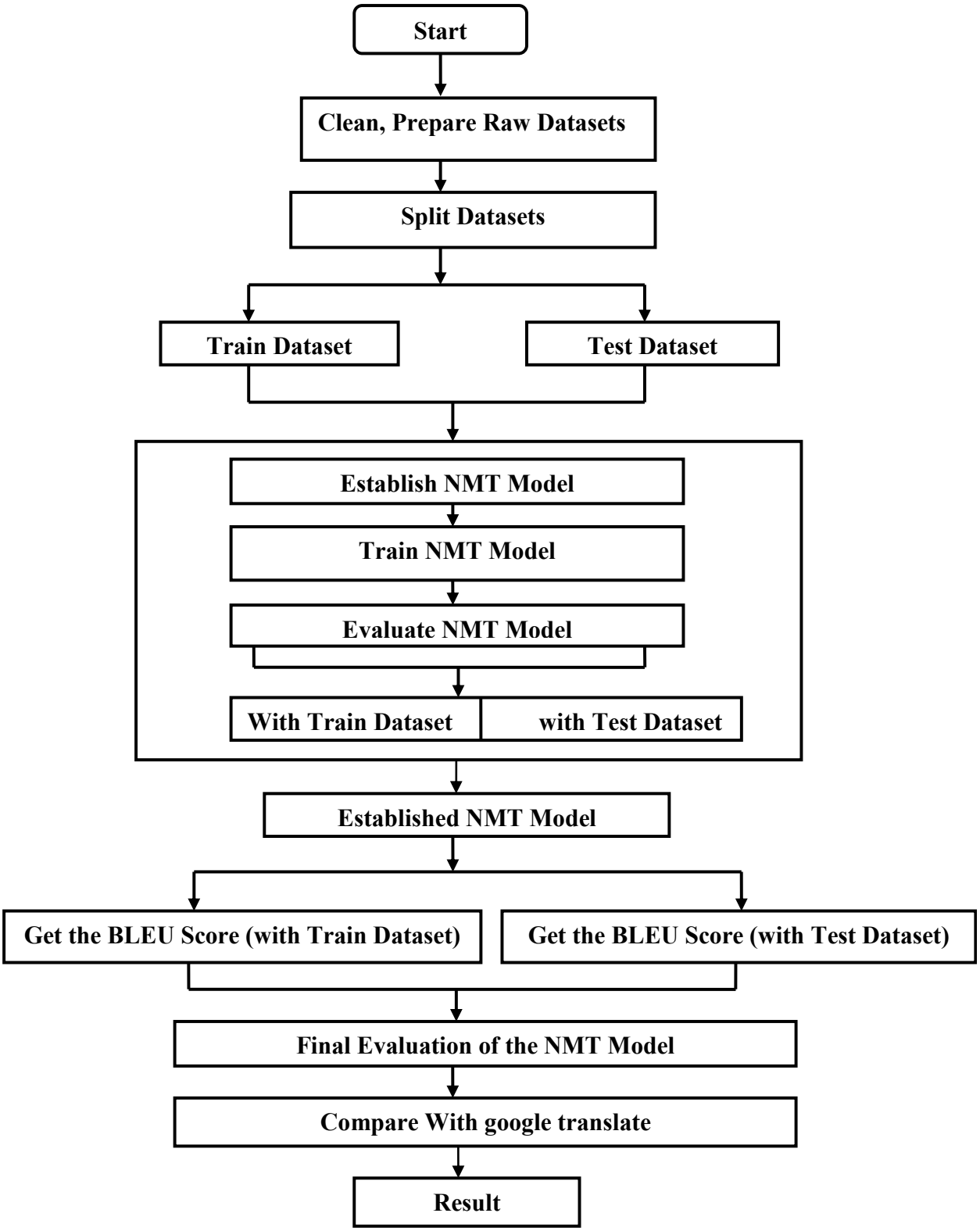


Figure 3.1: Work Flow Diagram of overall process

3.3 NEURAL MACHINE TRANSLATION

Neural Machine Translation is a machine translation approach that applies a large artificial neural network toward predicting the possibility of a sequence of words, often in the form of whole sentences. Unlike statistical machine translation, which consumes more memory and time, neural machine translation, NMT, trains its parts end-to-end to maximize performance. NMT systems are quickly moving to the front line of machine translation, recently outcompeting traditional forms of translation systems. Unlike the traditional phrase-based translation system which consists of many small sub-components that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation. Although neural machine translation (NMT) is a new approach, it is seen as a great breakthrough and has already become very popular among MT researchers, since it is clear that it improves the translation in most cases, offering an output that looks more fluid and more human. They say that NMT creates more fluent translations and can reduce post-editing efforts by up to 25%. For some linguistic professionals, there is also no doubt anymore that neural machine translation is performing better than rule-based or statistical machine translation. NMT systems understand and see the similarity of words, consider entire sentences and learn complex relationships between languages. The NMT uses a recurrent neural network, also called an encoder, to process a source sentence into vectors for a second recurrent neural network, called the decoder, to predict words in the target language. Now the most popular machine-translation method is neural machine translation due to its speed and accuracy. In phrase-based machine translation, there are many parts before translation. In the process, the sentence should be grouped in a phrase and then the system translates each phrase. After translating it has to reorder the words. Then the translation result have found. But on the other hand in neural machine translation, there are no such types of process. The sentence is given and the neural machine translation model translates the sentence to the targeted languages. The process is easier than the phrase-based machine translation system.

Comparisons between Phrase-based machine translation and neural machine translation have shown in figure 3.2.

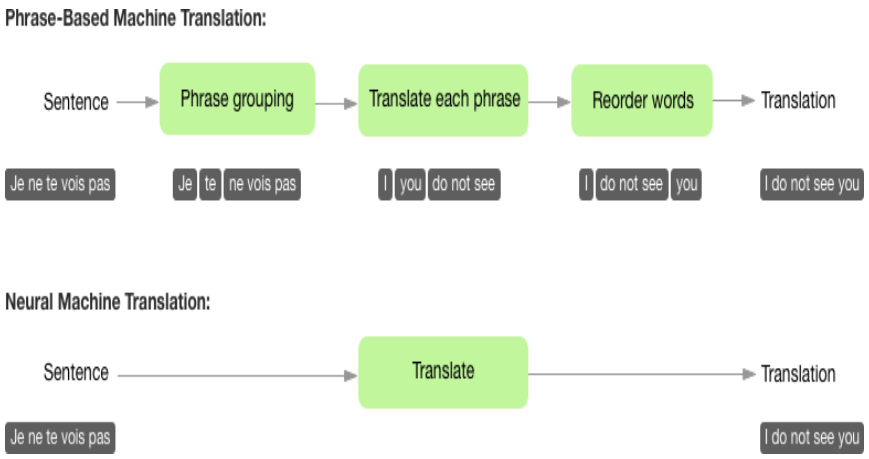
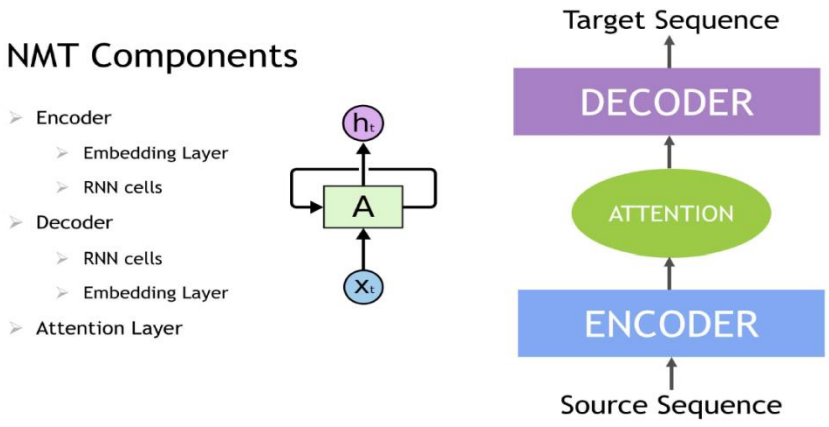


Figure 3.2: Comparison between Neural Machine Translation and Phrase-Based Machine Translation

.

Another figure 3.2 shows the neural machine translation components



.

Figure 3.3: Neural machine translation components

Here in the components there are encoder, decoder and attention layer. In an encoder and decoder, there is an embedding layer, recurrent neural network cells. In embedding layers, there is used word embedding. Word embedding is a technique to learn from text data. The attention layer is used for multiple language translation. There are some problems associated with neural machine translation. By using the attention layer problems have been solved. The neural translation model is one type of encoder-decoder model. The model encodes the input and decodes it in the output to get the target result. Encoder-Decoder consists of two recurrent neural networks (RNN) that act as an encoder and a decoder pair. The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence. An encoder neural network reads and encodes a source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole encoder-decoder system is jointly trained to maximize the chance of a correct translation.In deep learning, when it comes to NMT, a neural network architecture known as a recurrent neural network to model language sequences is used. Here in this research, a recurrent neural network named Long Short Term memory is used. The LSTM is mostly used for language translation. In figure 3.4 have shown the Encoder –decoder model.

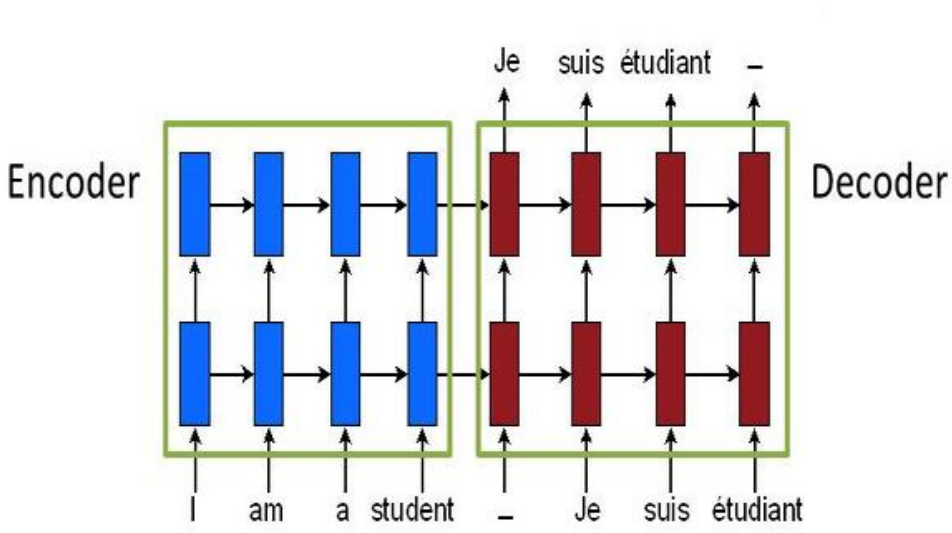


Figure 3.4: Encoder-Decoder Model

Encoder-decoder model consists of 3 parts: encoder, intermediate vector and decoder. The encoder accepts a single element of the input sequence at each time step, process it, collects information for that element and propagates it forward. The intermediate vector is the final internal state produced from the encoder part of the model. The model contains information about the entire input sequence to help the decoder make accurate predictions. Decoder predicts an output at each time step. Sequence-to-sequence prediction problems are challenging because the number of items in the input and output sequences can vary. For example, text translation and learning to execute programs are examples of seq2seq problems. Sequential is the easiest way to build a model in Keras. It allows building a model layer by layer. Each layer has weights that correspond to the layer the follows it. There use the add() function to add layers to the model. Neural Translation that is used here is a type of sequential model. Recurrent Neural Networks is a popular algorithm used in sequence models. Here both the input and output are sequences of data. Sequence Modeling is the task of predicting what word or letter comes next. Unlike the FNN and CNN, in sequence modelling, the current output is dependent on the previous input and the length of the input is not fixed. Sequence to Sequence models is a special class of Recurrent Neural Network architectures typically used to solve complex Language related problems like Machine Translation, Question Answering, creating Chatbots, Text Summarization, etc. So, this neural machine translation model is also a sequential model.

Word Embedding: Word Embedding is a Collective term for models that learned to map a set of words or phrases in a vocabulary to vectors of numerical values. Neural Networks are designed to learn from numerical data. Word Embedding is all about improving the ability of networks to learn from text data. Word embedding is a term for models that learned to map a set of words or phrases in a vocabulary to vectors of numerical values. Here in machine translation, a word embedding model is used for encoding the input portion of the languages. A word embedding is a learned representation for text where words that have the same meaning have a similar representation. There are various word embedding techniques that map (embed) a word into a fixed-length vector.

3.3.1 Long Short-Term Memory

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four layers, interacting in a very special way.In figure 3.5 have shown LSTM architecture.

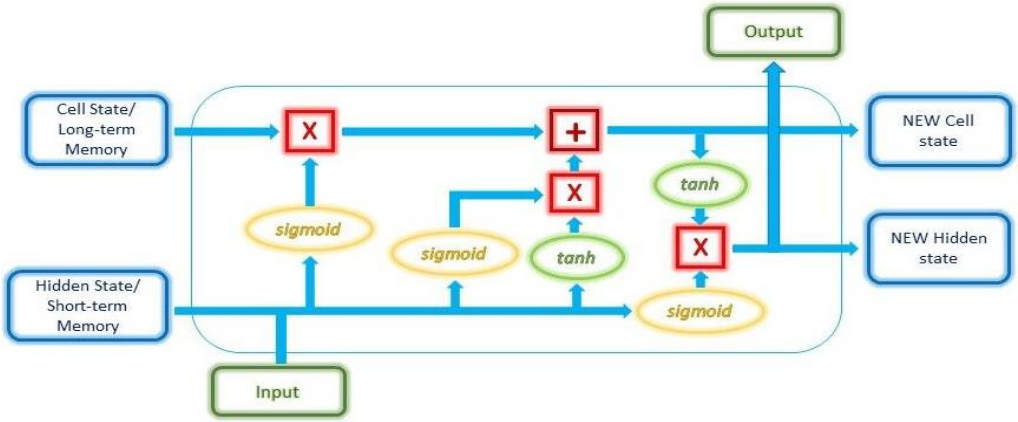


Figure 3.5: LSTM Architecture

From figure 3.5 have shown that LSTM has used two types of the activation function. One is Sigmoid and another is tanh. The sigmoid activation function is also called the logistic function. The function takes any real value as input and outputs in the range 0 to 1. The larger the input, the closer the output value will be to 1. Whereas the smaller the input, the closer the output will be to 0. The hyperbolic tangent activation function is also referred the tanh function. The function takes any real value as input and outputs in the range -1 to 1. The larger the input, the closer the output value will be to 1.0, whereas the smaller the input, the closer the output will be to 1.0. LSTM has a slightly more complex structure. At each time step, the LSTM cell takes in 3 different pieces of information - the current input data, the short-term memory from the previous cell and t he long-term memory. Short-term memory is commonly known as the hidden state, and long-term memory is usually known as the cell state. The cell then uses gates to regulate

the information to be retained or rejected at each time step. These gates need to be trained correctly to filter what is useful and what is not. These gates are called the Input Gate, the Forget Gate, and the Output Gate.

Input gate: The input gate decides what new information will be stored in the long-term memory. The gate only works with the information from the current input and the short-term memory from the previous time step. Therefore, gate filters out the information that is not useful. Mathematically, this is achieved using 2 layers. The first layer can be seen as the filter which selects what information can pass through it and what information to be rejected. To create this layer, the short-term memory and current input is passed into a sigmoid function. The sigmoid function will transform the values to be between 0 and 1. 0 indicating that part of the information is unimportant and 1 indicates that the information is important. The second layer takes the short term memory and current input as well and passes it through an activation function. The tanh function is used to manage the network. The outputs from these 2 layers are multiplied, and the final result represents the information. It is then used as output.

Forget Gate: The forget gate decides which information from the long-term memory should be retained or rejected. This is done by multiplying the incoming long-term memory by a forget vector generated by the current input and incoming short-term memory. The forget vector is also a particular filter layer. To obtain the forget vector, the short-term memory, and current input is passed through a sigmoid function.The forget gate is similar to the first layer in the input gate. But here a different weight is used. The outputs from the Input gate and the Forget gate will encounter a pointwise addition to give a new version of the long-term memory. This new long-term memory will also be used in the Output gate.

Output Gate: The output gate will take the current input and the new long-term memory to produce the new short-term memory. First, the previous short-term memory and current input will be passed into a sigmoid function with different weights. Then, the new long-term memory through a tanh function is settled. The output from these 2 processes will be multiplied to produce the new short-term memory. The short-term and long-term memory produced by these gates will then be carried over to the next cell for the process to be repeated. The output of each time step can be acquired from the short-term memory. The short term memory is also known as a hidden state.

RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. LSTM can by default retain the information for a long period. LSTM is used for processing, predicting and classifying based on time-series data. Here for this research, a neural machine translation model is established by using long short term memory. For this reason, LSTM has a profound contribution to this research. According to the results, both the GRU and LSTM were significantly better than the Simple RNN, but the LSTM was generally better overall. In figure 3.6 have shown architecture for translation language.

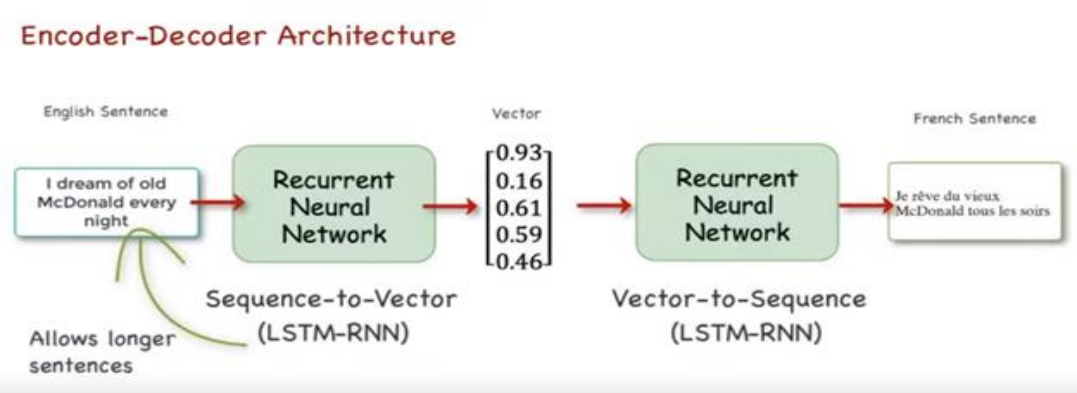


Figure 3.6: Language Translation by using LSTM-RNN

In this figure 3.6 have shown that by using LSTM-RNN the architecture translated one sentence from English to French. The sentence has gone through the Recurrent Neural Network and then turned into a vector. The vector representation has gone through the second Recurrent Neural Network and finally, the targeted translation have found. In our research, the translation from a different language to English has happened in this way.

3.3.2 Optimization Algorithm

The most popular is Adam. However, some people like to use a plain SGD optimizer with custom parameters. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based on training data. Adam is the combination of the advantages of two other

extensions of stochastic gradient descent. Adaptive Gradient Algorithm (AdaGrad) that maintains a per-parameter learning rate that improves performance on problems with sparse gradients.Natural language and computer vision problems is that kinds of problems. Root Mean Square Propagation (RMSProp) also maintains per-parameter learning rates that are charged based on the average of recent magnitudes of the gradients for the weight. This means the algorithm does well on online and non-stationary problems. Adam realizes the benefits of instead of adapting the parameter learning rates based on the average first moment as in RMSProp , Adam also makes use of the average of the second moments of the gradients. Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters beta1 and beta2 control the decay rates of these moving averages. Optimization is a type of searching process that can think of this search as learning. The efficiency of different optimization algorithm including Adam is shown in figure 3.7 below.

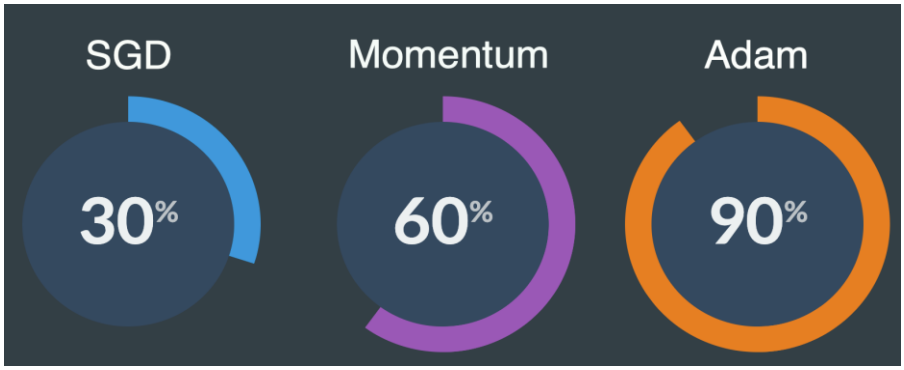


Figure 3.7: Comparison among SGD, Momentum, Adam

Here Adam has given the best result among other optimization algorithms. We used an optimization algorithm in the process to minimize the error of the model. Go with code then this portion defines the Neural Machine Translation Model where the model adds LSTM (Long Short Term Memory), Embedding layer, repeat Vector. Also, the model is a sequential model.

.

```
# define NMT model
def define_model(src_vocab, tar_vocab, src_timesteps, tar_timesteps, n_units):
model = Sequential()
model.add(Embedding(src_vocab, n_units, input_length=src_timesteps, mask_zero=Tru
e))
```

```
model.add(LSTM(n_units))
model.add(RepeatVector(tar_timesteps))
model.add(LSTM(n_units, return_sequences=True))
model.add(TimeDistributed(Dense(tar_vocab, activation='softmax')))
return model
```

Activation Function: Softmax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector. Softmax is an activation function. Softmax is exponential and enlarges differences – push one result closer to 1 while another closer to 0. Cross entropy is often computed for the output of softmax and true labels (encoded in one-hot encoding). By using a neural network several types of activation function are used for weight updating. Some of them are Binary Step, Sigmoid, Tanh, ReLU, Leaky-ReLU, Swis, and Softmax etc. An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network. An activation function is a very important feature of an artificial neural network. They decide whether the neuron should be activated or not. The softmax function is often described as a combination of multiple sigmoids. Sigmoid returns values between 0 and 1, which can be treated as probabilities of a data point belonging to a particular class. Thus sigmoid is widely used for binary classification problems. The softmax function can be used for multiclass classification problems. This function returns the probability for data point belonging to each class. Here is the mathematical expression of the softmax activation function

$$\sigma(\mathbf{z})_j = \frac{e_j^z}{\sum_{k=1}^{K} e_k^z} \quad \text{for } j = 1,\ldots K \qquad (3.1)$$

While building a network for a multiclass problem, the output layer would have as many neurons as the number of classes in the target. For example, if there are three classes, there would be three neurons in the output layer. Language translation is one types of multiclass classification problem. So, the activation function Softmax is the best.

3.4 DATASETS DETAILS

The researches have used Tab-delimited Bilingual Sentence pairs, http://www.manythings.org/anki/ for datasets. As typical neural language models rely on a vector representation for each word, a fixed vocabulary for both languages are used. Every out-of-vocabulary word was replaced with a special “UNK” token. 27 datasets are used for 27 different languages. Some of them are Asian languages and some of them are western languages. Among them 14 languages are mainly worked for this research. Mostly, it can be told that our research used both Latin script and non-script languages. The western languages are Romanian, Danish, Czech, Low Saxon, Icelandic, Macedonian and Low German. The Asian languages are Bengali, Thai, Marathi, Arabic, Cantonese, Tagalog and Vietnamese. Also, there used four languages that we cannot find in Google translator for translation. These four languages are Tagalog, Low German, Kabyle and Cantonese. To translate from these languages to English is not available in Google translator. Different datasets are used based on their availability. There is a used one-way translation. Different datasets are used to convert source languages to English languages. Datasets are one of the most important parts of this research. The higher sources of datasets give better accuracy in the translation. For poor resources of datasets the accuracy is also poor and also the translation quality is not satisfactory. Here used the different datasets for converting to source languages to English. For translating Arabic to English, Marathi to English, Thai to English, Danish to English, Czech to English every time different datasets are used according to the resources. So, in every translation from another language to English, required datasets have used the. To establish the translation process a huge amount of language pair and various datasets are used in this research.

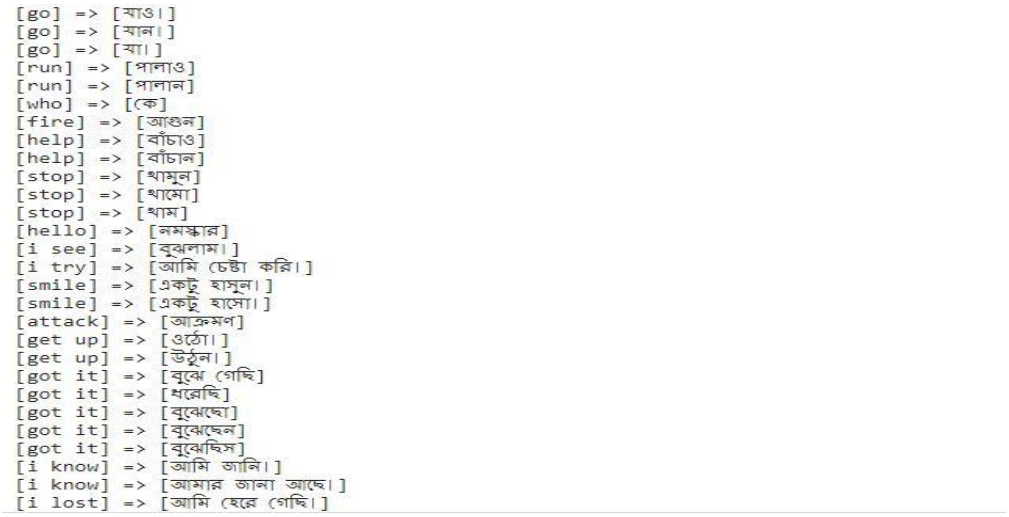


Figure 3.8: Preview of datasets

In figure 3.9 have shown the Preview Bengali-English datasets. After the preprocessing, the data of the datasets will be like that. For the Asian language that is mostly used non-Latin scripts. It is a difficult for machine to understand and translate the Latin scripts to English. Also in Google translate the accuracy of Asian languages is not as good as in western languages that use Latin scripts. So, most western languages use Latin scripts so for this reason, the accuracy or translation quality from western languages to English is better than the Asian languages. Every language used 100,200,300,400 epochs for training. Every dataset is split into training datasets and testing datasets. For training, a large portion of data have used. But in testing the volume of data are smaller than the training data. The amount of data in the phase of data processing can be increased or decreased. The model has used one portion of datasets for training and then used another dataset for testing. Some datasets have more sentence pairs than the others. The more trained the model with huge sentence pairs the better it can translate the languages in English. Below there have shown the table also mentioned the number of sentences every dataset has used. Table 3.1 have shown 10 languages datasets that have greatly analyzed in our research.

Table 3.1: Language Dataset for the 10 languages

Languages	Full Dataset	Training dataset	Testing dataset
Bengali	4332	3332	1000
Marathi	20000	19000	1000
Thai	2910	2000	910
Arabic	11584	9000	1000
Danish	10000	9000	1000
Romanian	10000	9000	1000
Czech	10000	9000	1000
Vietnam	6139	5000	1139
Icelandic	6558	6000	558
Macedonian	30000	29000	1000

Here in Table 3.1 have seen that not all languages the same datasets. The popular or most spoken languages have more sentence pairs than the other languages. For example here Marathi to English translation dataset contains 40751 sentence pairs where 20000 sentence examples are used for translation in our research. On the other hand; the Thai to English translation dataset contains 2910 sentence pairs which are far fewer comparing with the Marathi to English translation datasets. Asian language datasets are not as rich as western language datasets. Although Vietnam is a south Asian language it uses Latin scripts that are almost similar to the English languages. So, the translation quality is pretty good for the Vietnamese language. The Low German, Icelandic, Macedonian languages are western. But the translation quality of these languages needs to improve further. So, research has done with these languages for improving translation quality.

Datasets for the not available languages: As mentioned before 4 uncommon languages are used here for English translation. These languages translation into English is not available in the popular translator platform. So, we have started and worked further with these languages. The datasets for these languages have given below in Table 3.2.

Table 3.2: Language Dataset for the uncommon languages

Languages	Full Dataset	Training dataset	Testing dataset
Kablye	20000	18000	2000
Low German	3206	3000	206
Tagalog	3665	3000	665
Cantonese	3255	3000	255

Due to uncommon languages, the datasets are not available for these languages. These languages are Kabyle, Low Saxon, Tagalog and Cantonese. Among these languages let alone the Kabyle language other three languages datasets are not rich. For example, Low German has 3206 language pairs in the datasets. So, we used 3000 pairs for training and another 206 pairs are used for testing. The same things go for the Tagalog and Cantonese languages. The Tagalog language has full datasets consisted 3665 sentence pairs. The Cantonese language has 3255 sentence pairs. More than half of the datasets are used for training and the rest of the portion is used for testing. Among them Kablye languages have rich datasets consisted of 15000 sentence pairs.

Datasets for the common languages: The reaserch have been done further with 14 languages mainly. But let aside that to prove the efficiency of our model, there have also used 13 different languages datasets. Among most of them are western languages. They give a good translation quality is most of all language translator platform. The reasons for using these popular languages are to completely manifest the ability of our model. Table 3.3 has given the list of the languages datasets. It has mentioned the training and testing datasets below.Among the languages, we have seen that in most of the languages 10000 sentence pair's full datasets have used. Then 9000 sentence pairs have utilized for training and the rest 1000 sentence pairs for testing. But for the languages, Indonesian, Azerbaijani and Norwegian the full datasets are 7141, 2192 and 6015. For the training for the Indonesian languages, 6000 sentence pairs have taken and the rest of the 1141 sentence pairs are for testing. And for the Azerbaijani, the training dataset has 2000 sentence pairs and 192 for testing. At last for the Norwegian language, the training dataset is 5000 and testing datasets contain 1015 sentence pairs. The research has not done deeply with these languages. Because the main concern of our research is to improve the translation quality of the languages that are difficult to translate in English. But the process, methodology and implementation are the same as the other languages. The research has happenend deeply with other languages which need to improve translation quality.

Table 3.3: Language Dataset for the common languages

Language	Full Dataset	Train Dataset	Test Dataset
Turkish	10000	9000	1000
German	10000	9000	1000
Swedish	10000	9000	1000
French	10000	9000	1000
Portuguese	10000	9000	1000
Spanish	10000	9000	1000
Japanese	10000	9000	1000
Greek	10000	9000	1000
Bulgarian	10000	9000	1000
Finish	10000	9000	1000
Indonesian	7141	6000	1141
Azerbaijani	2192	2000	192
Norwegian	6015	5000	1015

3.4 ACCURACY TESTING

BLEU (Bilingual Evaluation Understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human. The closer a machine translation is to a professional human translation, the better it is. BLEU is one of the first metrics to claim a high correlation with human judgments of quality and remains one of the most popular automated and inexpensive metrics. The Bilingual Evaluation Understudy Score, or BLEU for short, is a metric for evaluating a generated sentence to a reference sentence. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0. The score was developed for evaluating the predictions made by automatic machine translation systems [59]. It is not perfect, but does offer 5 compelling benefits:

i. Quick and inexpensive to calculate.
ii. Easy to understand.
iii. Language independent.
iv. High correlation with human evaluation.

BLEU measures the overlap of unigrams (single words) and high-order n-grams between MT output and reference translations [60]. The main component of BLEU is n-gram precision. n-gram precision is calculated as the ratio between matched n-grams and the total number of n-grams in the evaluated translation. Precision is calculated separately for each n-gram order, and the precisions are combined via a geometric averaging. The highest n-gram order is defined commonly to be four. Higher-order n-grams are used as an indirect measure of a translations level of grammatical well-turned. The BLEU metric computes the modified precision score, weighted by the brevity penalty, which penalizes sentences that are shorter than the reference. BLEU is typically computed over the entire corpus, not single sentences. Very few translations will attain a score of 1 unless they are identical to a reference translation. For this reason, even a human translator will not necessarily score 1, as there is great variability of possible correct translations. In this sense, having more reference translations per sentence is highly welcome. Both metrics, BLEU and NIST, focus only on n-gram precision and disregard recall. The recall is the ratio between matched n-grams and the total number of n-grams in the reference translation.

Comparison: There are many types of the machine translation system. But among them, neural machine translation got the best result. Before Google translate used statistical machine translation. But neural machine translation is used since 2016. The research has used neural machine translation. Comparing with the Google translate results are better in our proposed system. So, three types of comparison have been done. First has happened in the training and the testing phase. In this phrase, The Target and sentences predicted by the model are compared. The BLEU score of our model with the BLEU score of Google translator have compared. At last, comparing between sentences is showed. Significant result have found in the comparison. For every language, there have used the same sentences and analyze the translation. Then the output of the established model and the output that is found in the Google translator has compared. So, for comparison, Google translate is used which is now the most popular among the translator system.

3.6 SUMMARY

There discussed the algorithm, model, optimization algorithm that are used in this research. Although a neural translation model is used here some other algorithm and model is also used. For evaluating the translation the most popular algorithm named BLEU that are also discussed. In short, the methods that are used in this research have been discussed briefly in this section.

CHAPTER 4

IMPLEMENTATION

4.1 INTRODUCTION

Here the Google co-lab is used for implementation. Different types of python packages are used for this. Tensorflow, Keras, tokenizer, pickle, NLTK etc are used for translating from a different language to English. Mainly four steps are used for implementation. These are data preparation, train and establish the model. The last step is to evaluate the established neural translation model. These steps are described elaborately in this chapter.

4.2 DATA PREPARATION

First, here have worked with the datasets that are used for translation. Different datasets are used for different languages.The research cannot be done with the raw datasets. So, first, the dataset preparation has happened. The preparation of datasets is divided into two sections. These are described below.

4.1.1 Text Cleaning

Text cleaning means preparing the raw datasets for the model to translate from one language to another. When the sentences located in datasets normally the dataset is not processed. For translation, read and processed the data is must. For translation, datasets have used that are contained sentences pairs for different languages. For this reason, the first step is to clean the text. The function called load_doc() will load the file as a blob of text. The term blob stands for Binary Large Object. blob is used for storing

information in databases. Then the organization to clean each sentence is happened. The specific cleaning operations that have performed to clean text are as follows:

i. Remove all non-printable characters.
ii. Remove all punctuation characters.
iii. Normalize the case to lowercase (For non-Latin characters it does not perform).
iv. Remove any remaining tokens that are not alphabetic.

These operations have performed on each phrase for each pair in the loaded dataset**.** First, we have to load the datasets to the memory. Sentences are divided in the file. Then, we have to work with remove punctuation, another unwanted or duplicate sentence. Then sentences are broken into pair according to source-English sentences. Different types of languages are used to translate into English. In short, the whole process has been shown below in figure 4.1.

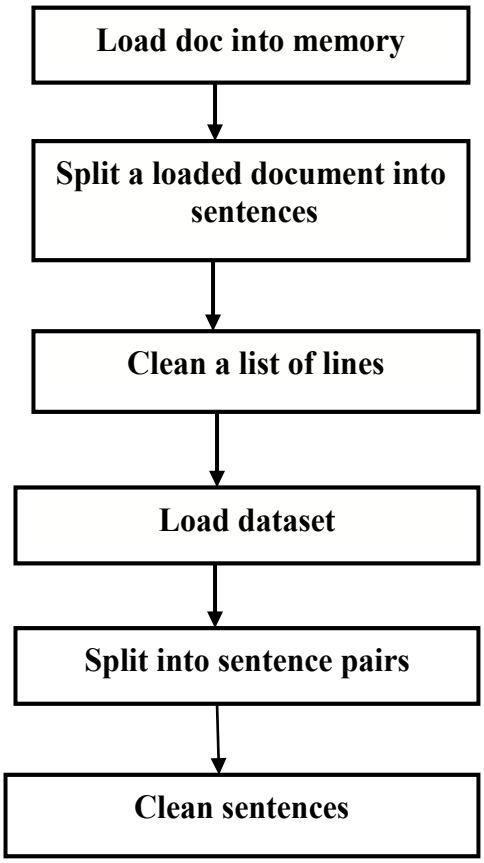


Figure 4.1: Flow chart of preparing the text

4.1.2 Text Splitting

This is a good number of examples for developing a small translation model. The complexity of the model increases with the number of examples, length of phrases, and size of the vocabulary. We have a good dataset for translation. Most of the western languages there used 10000 sentences datasets. But for the Asian and rare languages, the datasets are small compare to other languages. The splitting of text happened in this way. For example, there are 4332 sentences full in Bengali to English Datasets. Further, we will then stake the first 3,332 of those as examples for training and the remaining 1,000 examples to test the fit model. In this way, we have to split the datasets into train datasets and test datasets. Train dataset is used to train the model and the test dataset is used to test the model. Training Dataset used the sample of data used to fit the model. Test datasets used to provide an unbiased evaluation of the final model fit on the training datasets. We can fix the dataset size. Sometimes for some languages datasets, the whole dataset is not used. Some Portion of the dataset is used. So, according to this, fixing the dataset size and split the dataset is an important part of the data processing or preparation process. In figure 4.2 there have shown the whole process in a flow chart.

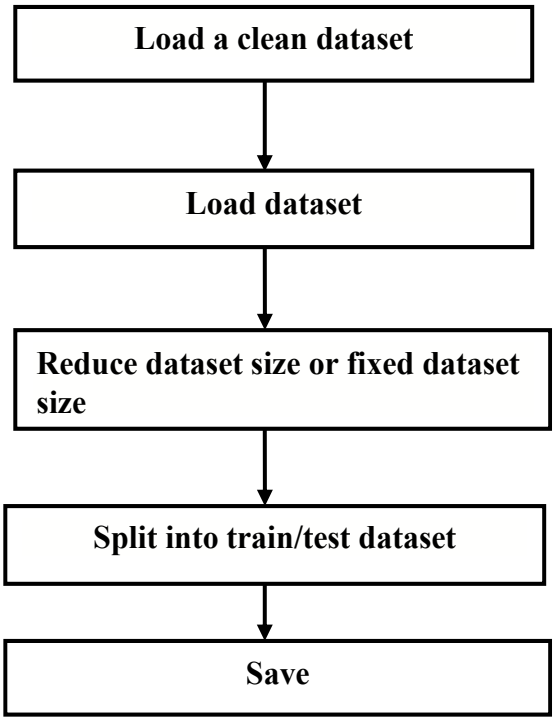


Figure 4.2: Flow chart of split text

4.3 TRAIN AND ESTABLISH NMT MODEL

The next step is to train the model. For translating from one language to another, every sentence is tokenized and then the maximum vocabulary length of both source and target language is calculated. The basic form of tokenization is to split the words in a sentence with extra space between words in the sentence. Keras Tokenize class is used to map words to integers, as needed for modeling. A separate tokenizer for the source sequences and the target sentences is also used. The function named create_tokenizer() trained a tokenizer on a list of phrases sequences. Similarly, the function named max_length() below find the length of the longest sequence in a list of phrases.These functions with the combined dataset is used to prepare tokenizers, vocabulary sizes, and maximum lengths for both the source and target sequences. For neural machine translation, one of the most important things is to encode and decode. Encoder-Decoder captures both semantic and syntactic structures of the phrases.Word Embedding is used to encode the source sentence.In our research one hot encode have applied for the target sequence. Each bit represents a possible category. If the variable cannot belong to multiple categories at once, then only one bit in the group is existed. This is called one-hot encoding. The one-hot encoding creates one binary variable for each category. A one-hot encoding is appropriate for categorical data where no relationship exists between categories. Each input and output sequence must be encoded to integers and padded to the maximum phrase length. This is because a word embedding used for the input sequences and one hot encodes the output sequences. The function named encode_sequences() perform these operations and return the result. Another reason used one-hot encoding is the model will predict the probability of each word in the vocabulary as output. The function encode_output() do one-hot encode English output sequences. One hot encode transforms all the sequences into binary vectors. After doing all of this now the model is defined. An encoder-decoder LSTM model is applied on this problem. In this architecture, the input sequence is encoded by a front-end model called the encoder then decoded word by word by a backend model called the decoder. The function define_model()defines the model and takes several arguments used to configure the model, such as the size of the input and output vocabularies, the maximum length of input and output phrases, and the number of memory units used to configure the model. The encoder reduces the whole sequence to a context of fixed length, and the

decoder heavily depends on the context from the encoder. Hence, the model finds it difficult to deal with longer sentences. For longer sentences, the initial context will be lost before the end of the given sequence. The encoder produces a 2-dimensional matrix of outputs, where the length is defined by the number of memory cells in the layer. The decoder is an LSTM layer that expects a 3D input of [samples, time steps, features] to produce a decoded sequence of some different length defined by the problem. These can be solved this using a RepeatVector layer. This layer simply repeats the provided 2D input multiple times to create a 3D output. The RepeatVector layer can be used as an adapter to fit the encoder and decoder parts of the network together. TimeDistributedDense applies the same dense fully-connected operation to every timestep of a 3D tensor. Another feature is called timestamps, which captures the order between sequences. The dimension in the embedding creates a matrix representation for each word. For example, an embedding dimension of 100, then each word will have a matrix of (1,100). A generic function is developed to define an encoder-decoder recurrent neural network. Masking is also used in the model. Masking is necessary for two things. First, masking prevents calculating a score for the blank space in the sequence. Each sentence is padded to prevent inputting a sequence of varying length to the model. The second reason no to see all the words at once.

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. Think of a batch as a for-loop iterating over one or more samples and making predictions. At the end of the batch, the predictions are compared to the expected output variables and an error is calculated. From this error, the update algorithm is used to improve the model. The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch is comprised of one or more batches. The number of epochs is traditionally large, often hundreds or thousands, allowing the learning algorithm to run until the error from the model has been sufficiently minimized. Creating a train and test split of our dataset is one method to quickly evaluate the performance of an algorithm on our problem. The training dataset is used to prepare a model, train it. The test dataset is new data where the output values are withheld from the algorithm. The trained model on the inputs from the test dataset gathered predictions and compare them to the withheld output values of

the test set. Comparing the predictions and withheld outputs on the test dataset allows us to compute a performance measure for the model on the test dataset. This is an estimate of the skill of the algorithm trained on the problem when making predictions on unseen data. The model is trained using the efficient Adam approach to stochastic gradient descent and minimizes the categorical loss function. The prediction problem has framed as multi-class classification. ADAM has given better than the SGD. So, here ADAM is used. The model configuration was not optimized for this problem, meaning that there are plenty of opportunities to tune it and lift the skill of the translations. We trained the model for 100,200,300,400 epochs for different times and a batch size of 64 is used. We tried different batch size also but this did not give a good result. Check points are checked to ensure that each time the model skill on the test set improves. The model is saved to file 'model.h5' for later evaluation or translation. The code for the fit model has given below.

```
# fit model
filename = 'model.h5'
checkpoint = ModelCheckpoint(filename, monitor='val_loss', verbose=1, save_best_onl
y=True, mode='min')
model.fit(trainX, trainY, epochs=100, batch_size=64, validation_data=(testX, testY), ca
llbacks=[checkpoint], verbose=2)
```

In the checkpoint, the validation loss has monitored and the mode is min. There have shown for the code for 100 epoches. We increase the epoch size. In figure 4.3 have shown the overall flow chart of the train and test the model.The model configuration was not optimized for this problem, meaning that there is plenty of opportunities for us to tune it and lift the skill of the translations

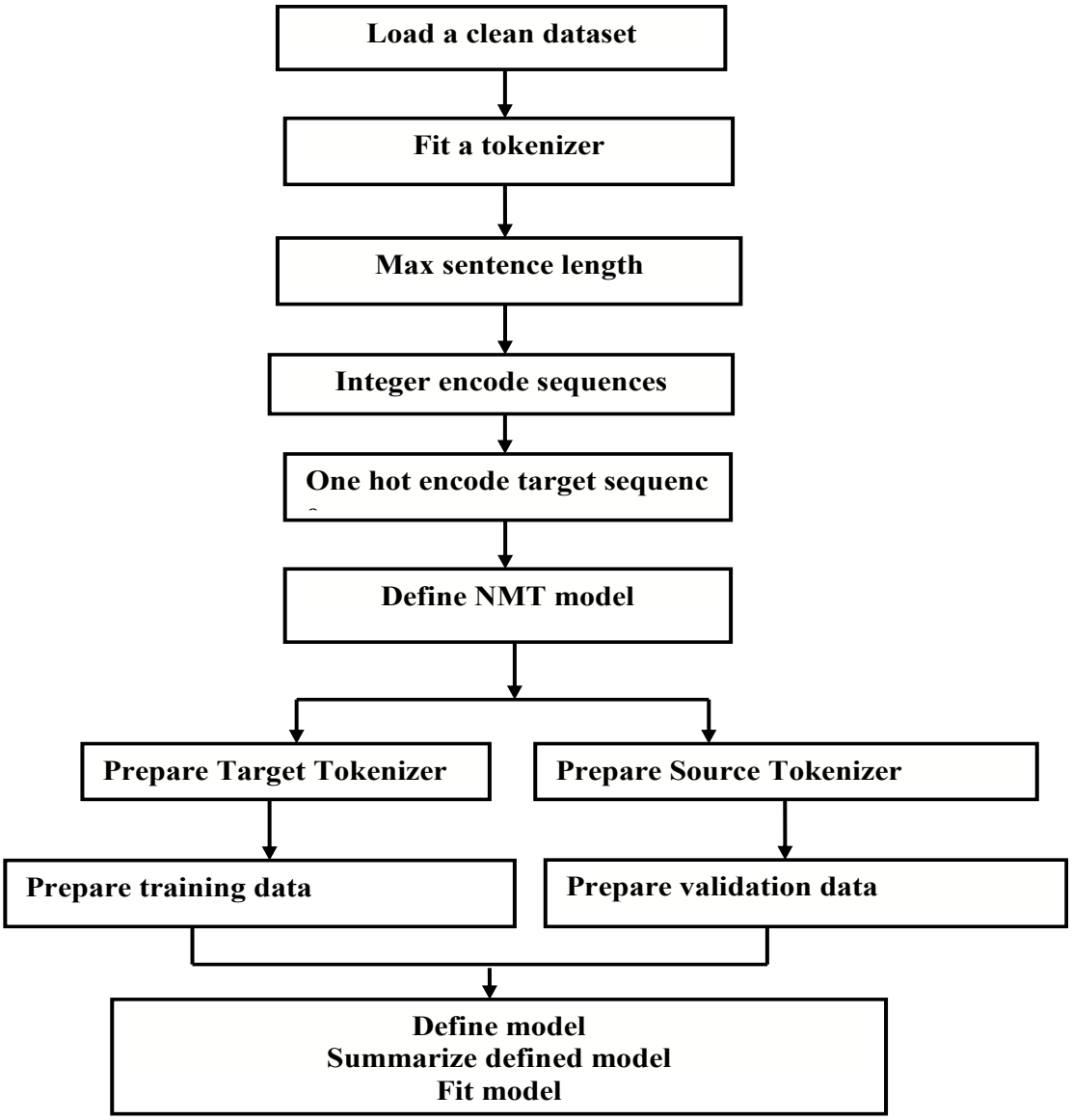


Figure 4.3: Flow chart of train and test the model

Running the example first prints a summary of the parameters of the dataset such as vocabulary size and maximum phrase lengths. A plot of the model is also created providing another perspective on the model configuration. **.** The plot model graph of the NMT and Epoch process has given below in figure 4.4 and 4.5.

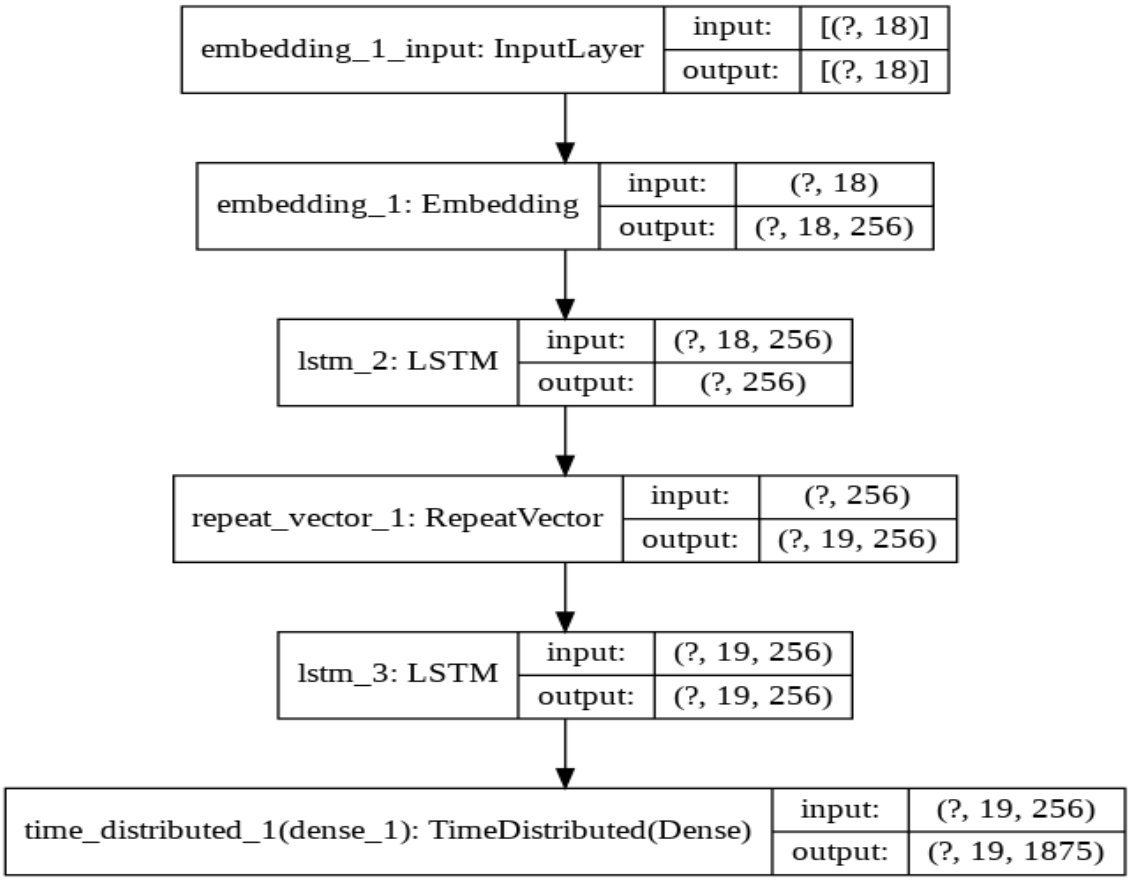


Figure 4.4: Plot of the model graph of NMT

Everytime when to train the model, this types of model graph have found which indicates the flow of how the model works.Next, the model is trained.Each epoch takes about 30 seconds on modern CPU hardware, no GPU is required.During the run Epoch phase will be like that.

Epoch 1/100

Epoch 00001: val_loss improved from inf to 1.81009, saving model to model.h5
68/68 - 33s - loss: 2.8035 - val_loss: 1.8101

Epoch 2/100

Epoch 00002: val_loss improved from 1.81009 to 1.60016, saving model to model.h5
68/68 - 32s - loss: 1.6904 - val_loss: 1.6002

Epoch 3/100
Epoch 00003: val_loss improved from 1.60016 to 1.51209, saving model to model.h5
68/68 - 33s - loss: 1.5745 - val_loss: 1.5121

Epoch 4/100
Epoch 00004: val_loss improved from 1.51209 to 1.44415, saving model to model.h5
68/68 - 33s - loss: 1.4857 - val_loss: 1.4441

Epoch 5/100
Epoch 00005: val_loss improved from 1.44415 to 1.41067, saving model to model.h5
68/68 - 33s - loss: 1.4407 - val_loss: 1.4107

Epoch 6/100
Epoch 00006: val_loss improved from 1.41067 to 1.40187, saving model to model.h5
68/68 - 33s - loss: 1.4131 - val_loss: 1.4019
Epoch 7/100…….

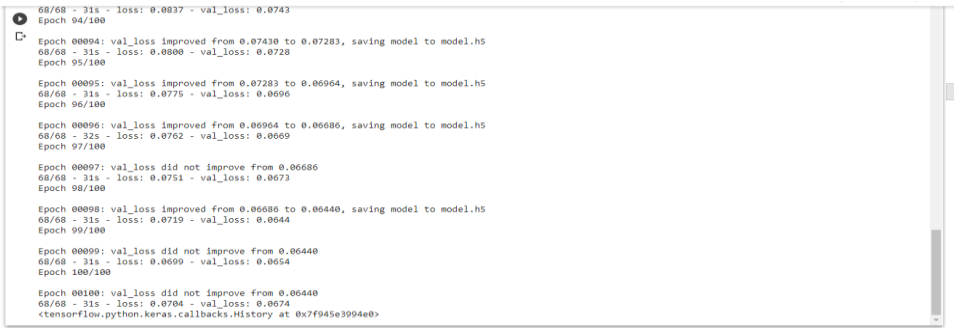


Figure 4.5: Train Neural machine Translation model

When we translate the model then during the training loss and validation loss summary will be like that. Loss and validation loss can be equal, greater than one another or less than one another. The impact of this and the result have found for established model has discussed in the result and discussion.

4.4 EVALUATE NEURAL TRANSLATION MODEL

The model is evaluated based on the train and the test dataset. The model should perform very well on the training dataset and ideally have been generalized to perform well on the test dataset. In figure 4.6 has shown the whole process for evaluating the model.Ideally, separate validation dataset use to help with model selection during training instead of the test set.

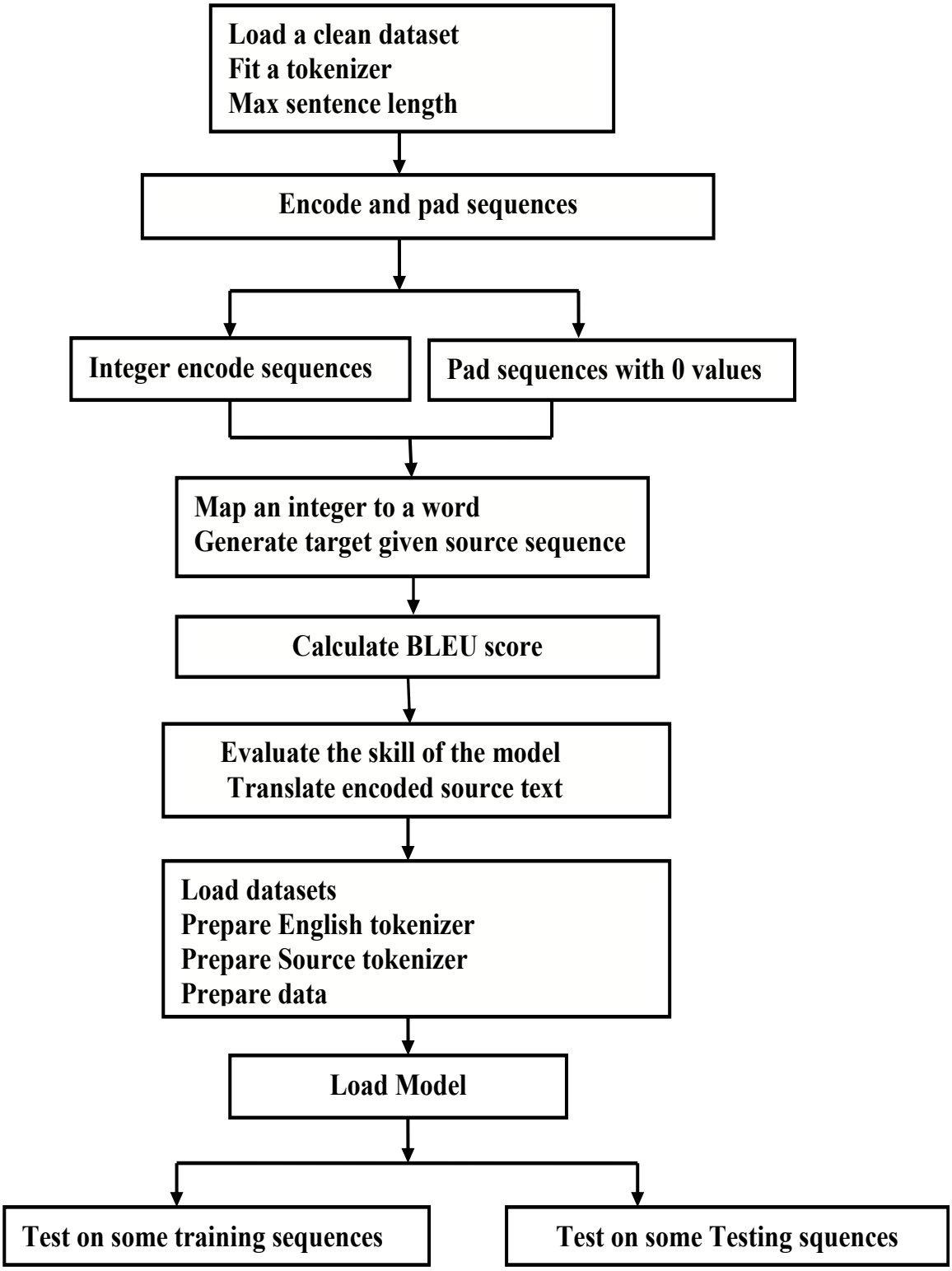


Figure 4.6: Flowchart for evaluating machine translation system

But here is used a test dataset for testing the evaluation. The clean datasets must be loaded and prepared as before. The good practice is to use separate datasets for the validation.In the first step, the clean dataset is loaded. By using the datasets the tokenizer is fitted and gets the maximum sentence length for both the source and destination sentence. Then encode and pad the input sequences is happened.The integer encode sequence has done by using text_to_sequences() function. This will be a sequence of integers that can enumerate and lookup in the tokenizer to map back to words. And the padding has happened with 0 values. Then every integer is mapped to a word. The function named word_for_id(), will perform this reverse mapping. This mapping can perform for each integer in the translation and return the result as a string of words. To generate the target sequences based on source sequences a function *predict_sequence()* have used. This is repeated for each source phrase in a dataset and compares the predicted result to the expected target phrase in English. All these functions are found in the Keras Packages of python. Now the predicted sequences have gained. Starting with the conclusion, the model can predict the entire output sequence in a one-shot manner. Now To evaluate the skill of the model the BLEU score is used. The BLEU score is calculated to get a numerical idea of how well the model has performed. The translation of encoded source text has happened during the evaluation of the model. Evaluation involves two steps: first generating a translated output sequence, and then repeating this process for many input examples. The skill of the model have summarized across multiple cases. The evaluate_model() function is worked for the evaluation of the model. The datasets have loaded for both training and testing. The clean datasets must be loaded and prepared as. The model will be evaluated by the train and the test dataset. The model should perform very well on the training dataset and ideally have been generalized to perform well on the test dataset. These datasets are used to judge the performance of the model. Previously the model has saved for further work. So, the best model saved during training must be loaded. Both the model and the datasets have to be loaded. Some training and testing sequences is tested. Some training or testing sentences have shown later in the result analysis and discussion chapter for every language. The BLEU score of training and testing datasets for the model is also found here. The model is trained again and again and every time two LSTM layer is used. Every LSTM contains four layers. Then in two LSTM layers, 8 layers have used in these model.

4.5 SUMMARY

The whole implementation process has been discussed here. The raw datasets has made into useful ones for using translation. Also, the model have trained and evaluated. Every step of how the training of the model happened has discussed. The whole chapter has described Functions, methods, algorithms, how many layers are used and every layer how it works associated with this research. The whole process of how the implementation has been being discussed here.

CHAPTER 5

RESULT ANALYSIS AND DISCUSSION

5.1 INTRODUCTION

This chapter here discussed the result of this research. Different languages have given different translation quality. But in this research 14 languages translation quality has measured and analyzed properly. First, the training result and the BLEU score of training datasets have shown. The training loss and validation loss of the model have also compared. Then the testing results of every language and also the BLEU score of testing datasets have indicated. In both training and testing, the target sentence is compared with the predicted sentence according to the source sentence. Then the next section the BLEU score of Google translator of different languages to translate in English have compared with our established model. Then different languages translation quality have manifested according to Google translator with samples of sentences by sentences. All the languages that are translated to English have discussed here. Point to be noted that in this research our research have worked with four languages for translating to English that has not even initiated in Google translator. At last the whole chapter have summarized in the summary.

5.2 TRAINING RESULTS AND ANALYSIS

14 different languages have shown which have used this neural translation model. In the first section, the overall model result for every language for translating into English have discussed. The target, predicted sentences for every language for the training session and the BLEU score indicated. In the second section, the testing result for 14 languages also the BLEU score of testing datasets have exhibited.

5.2.1 Bengali Language

Bengali is the official and national language of Bangladesh with 98% of Bangladesh is using Bengali as their first language. Bengali is the most widely spoken language of Bangladesh and the second most widely spoken of the 22 scheduled languages of India, after Hindi. With more or less 228 million native speakers and another 37 million second-language speakers, Bengali is the fifth most spoken native language. It is also the seventh most spoken language by the total number of speakers in the world. Before going through the training the established model has worked with vocabulary size and maximum length. The neural translation model is a sequential model all the work of the model has done sequentially. For the Bengali language, the vocabulary size is 3289 and the maximum length of the vocabulary is 17. On the other hand for output in English, the English vocabulary size is 1859 and the maximum length of the vocabulary is 15. For every time to train the model four layers are used. The first layer is the embedding layer the second is the LSTM layer, then a repeat vector is used and then another LSTM layer is utilized. Another thing is that for every LSTM layer inside this layer another four layers are used. And at last, the input goes through the time distributed layer. After go through these four layers finally the output has found. Here the total parameter is 2,370,371 and all parameters are trained through the model. The embedding size defines the length of the vectors used to represent words. A larger dimensionality will result in a more expressive representation, and in turn, better skill. Interestingly, the results show that the largest size tested did achieve the best results, but the benefit of increasing the size was minor overall. For the embedding layer, 256 sizes embedding layer have used.. For the first embedding layer, the numbers of parameters are 841984. After the first LSTM layer, 525312 parameters have found. By using two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the number of the parameters is the same. After going through the time distributed layer the numbers of parameters are 477763. The total parameter is 1104329 and all parameters are trained through the model.The summary of the total model result for the Bengali language is given below.

English Vocabulary Size: 1859

English Max Length: 15

Bangla Vocabulary Size: 3289

Bangla Max Length: 17

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 17, 256)	841984
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 15, 256)	0
lstm_1 (LSTM)	(None, 15, 256)	525312
time_distributed (TimeDistri	(None, 15, 1859)	477763

Total params: 2,370,371

Trainable params: 2,370,371

Non-trainable params: 0

Now the tables 5.1 and 5.2 have shown the training result and BLEU score of the Bengali language. In the training phase, the model knows the sentences properly. So, the training result is pretty good. Also in figure 4.1 have shown the comparison between training loss and validation loss. Both are decreasing and an expected training loss is lower than the validation loss.

Table 5.1: Source, Target and Predicted Sentences for Bengali Language (Train)

Source	Target	Predicted
উনি কথা বলছিলেন।	he was speaking	he was speaking
ওরা পরোয়া করে না।	they dont care	they dont care
টম মেরিকে হুমকি দিলো।	tom threatened mary	tom threatened mary
ফুলদানিটা কে ভাঙল|	who broke the vase	who broke the vase
টম বারে দাঁড়িয়ে বিয়ার খাচ্ছিল।	tom was standing at the bar having a beer	tom was standing at the bar having a beer
আমি বেকারিতে গেলাম।	i went to the bakery	i went to the bakery

Table 5.2: BLEU Score for Bengali Language (Train)

BLEU-1	0.946882
BLEU-2	0.923017
BLEU-3	0.893905
BLEU-4	0.822678

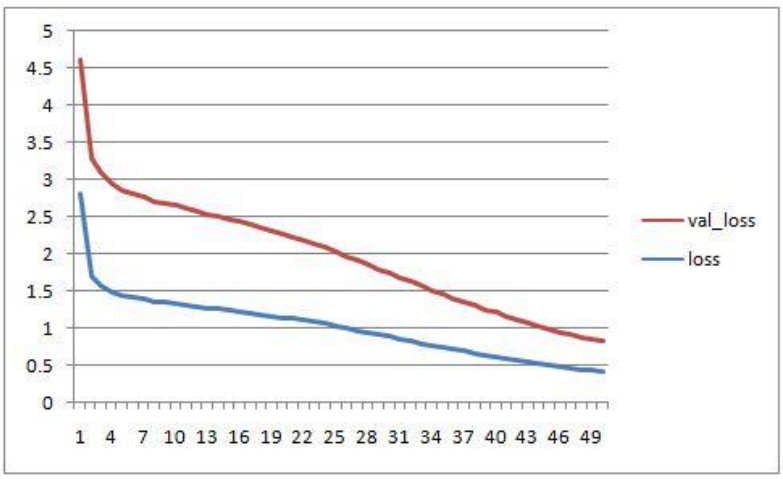


Figure 5.1: Comparison of loss and val_loss for Bengali language.

5.2.2 Romanian Language

The Romanian language is a Balkan Romance language. The language is spoken by around 24–26 million people as a native language, primarily in Romania and Moldova. Romanian keeps several features of Latin, such as noun cases, which have disappeared from other Romance languages. Here in this research, has given a better BLEU score for the Romanian language than Google translator. For every language, the first step is to find the vocabulary size and maximum length of the vocabulary. For the Romanian language, the vocabulary size is 6121 and the maximum length of the vocabulary is 14. On the other hand for output as the English language, the English vocabulary size is 4297 and the maximum length of the vocabulary is 10. For the first embedding layer, the numbers of parameters are 1566976. After the first LSTM layer, 525312 parameters have found. By using two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the number of the parameters is the same. After going through the time distributed layer the numbers of parameters are 508089. The total parameter is 1104329 and all parameters are trained through the model. The summary of the total model result for the Romanian language is given below.

English Vocabulary Size: 4297
English Max Length: 10
Romanian Vocabulary Size: 6121
Romanian Max Length: 14
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 14, 256)	1566976
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 10, 256)	0
lstm_1 (LSTM)	(None, 10, 256)	525312

_____

time_distributed (TimeDistri (None, 10, 4297) 1104329

==========

Total params: 3,721,929

Trainable params: 3,721,929

Non-trainable params: 0

Table 4.3 has shown the training result of the Romanian language. In table 4.3 the BLEU score of training datasets have shown. The comparison of training loss and validation loss for the Romanian language for this model has appeared in figure 5.2. The training loss and validation loss is almost the same which indicates the model works properly.

Table 5.3: Source, Target and Predicted Sentences for Romanian Language (Train)

Source	Target	Predicted
dansul meu favorit este tangoul	my favorite dance is the tango	my favorite dance is the tango
ai cheltuit multi bani	you spent a lot of money	you spent a lot of money
nuti pot spune cine a facut asta	i cant tell you who did that	i cant tell you who did that
mai facut sami pierd mintile	you made me lose my mind	you made me lose my mind
aceasta problema merita discutata	this problem is worth discussing	this problem is worth discussing
tom a plecat in graba	tom hurried away	tom hurried away
el este un pictor	he is a painter	he is a painter

Table 5.4: BLEU Score for Romanian language (Train)

BLEU-1	0.979510
BLEU-2	0.971432
BLEU-3	0.965954
BLEU-4	0.941844

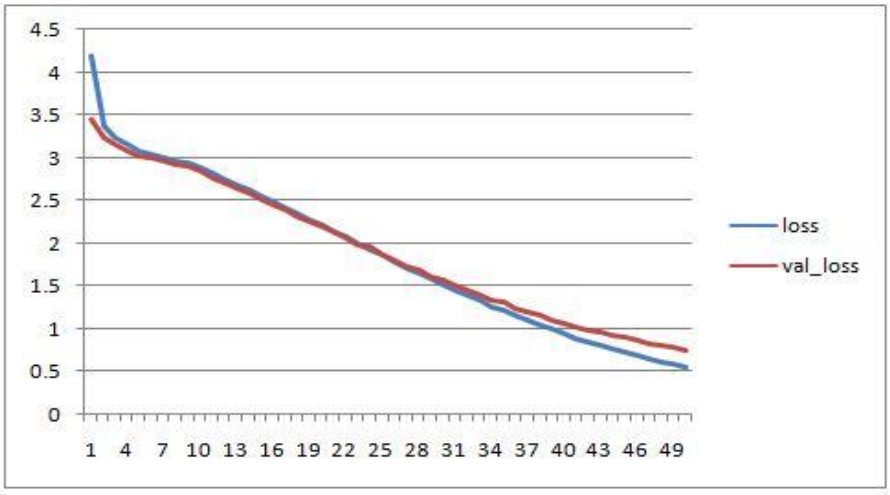


Figure 5.2: Comparison of loss and val_loss for Romanian language

5.2.3 Danish Language

Both Danish and English belong to the Germanic language family. English has much more similarity with Danish than any other language. The modern Danish alphabet is similar to the English one. The language uses Latin scripts. Danish is a North Germanic language spoken by about six million people, mostly in Denmark, Greenland.For the Danish language, the vocabulary size is 4246 and the maximum length of the vocabulary is 11 for our model. On the other hand for output as the English language, the English vocabulary size is 3297 and the maximum length of the vocabulary is 8. For the first embedding layer, 1086976 parameters have found. After the first LSTM layer, the numbers of the parameters are 525312. By using two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the numbers of the parameters are the same. After going through the time distributed layer, 847329 parameters have found. The total parameter is 2,984,929 and all parameters are trained through the model. The summary of the total model result for the Danish language is given below.

English Vocabulary Size: 3297
English Max Length: 8
Danish Vocabulary Size: 4246
Danish Max Length: 11

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 11, 256)	1086976
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 8, 256)	0
lstm_1 (LSTM)	(None, 8, 256)	525312
time_distributed (TimeDistri	(None, 8, 3297)	847329

Total params: 2,984,929

Trainable params: 2,984,929

Non-trainable params: 0

In table 5.5 have shown the comparison of Target and Predicted sentence according to the source sentence in the training session. Also, the BLEU score for the training datasets of this established model have shown in table 5.6.

Table 5.5: Source, Target and Predicted Sentences for Danish Language (Train)

Source	Target	Predicted
giv mig mine penge tilbage	give me my money back	give me my money back
tnk over det	think about it	think about it
lad mig vre i fred	leave me alone	leave me alone
det rager mig en papand	i couldnt care less	i couldnt care less
han kan ikke tage en beslutning	he cant make a decision	he cant make a decision
lad mig vide hvordan det gar	let me know how it goes	let me know how it goes
hvad lavede du om aftenen	what did you do at night	what did you do at night
aftensmaden er klar	dinner is ready to eat	dinners ready

Table 5.6: BLEU Score for Danish Language (Train)

BLEU-1	0.965638
BLEU-2	0.953126
BLEU-3	0.942366
BLEU-4	0.886245

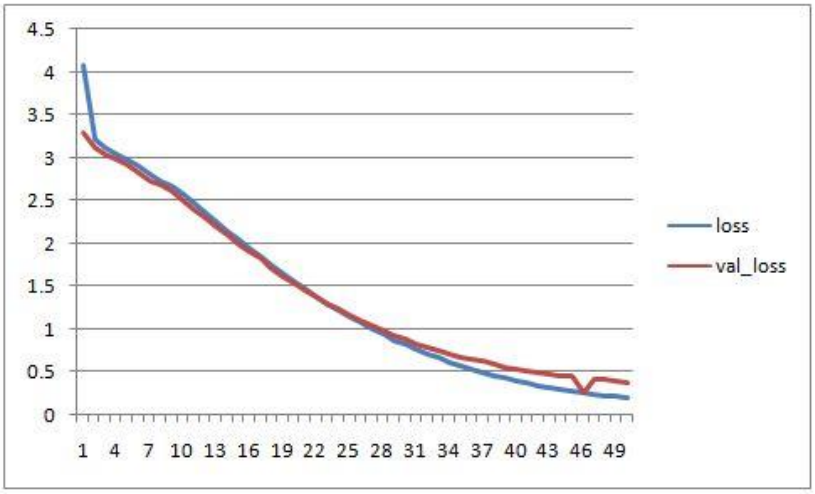


Figure 5.3: Comparison of loss and val_loss for Danish language

The comparison between the training loss and validation loss has shown in figure 5.3. This curve tells a lot about the behavior and characteristics of the neural translation model. It has also proved that how well the tuning has happened for this model.

5.2.4 Czech Language

The official language of the Czech Republic is Czech. The language is spoken by nearly 11 million native speakers; Czech is classified as part of the Slavic branch of Indo-European languages. Like other Slavic languages, Czech is a fusion language with a rich system of morphology and relatively flexible word order. For the Czech language,

the vocabulary size is 6536 and the maximum length of the vocabulary is 8. On the other hand for output as the English language, the English vocabulary size is 3257 and the maximum length of the vocabulary is 8. For the first embedding layer, the numbers of the parameters are 1673216.After the first LSTM layer, 525312 parameters have found. By using two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the number of the parameter are the same. After going through the time distributed layer the numbers of the parameters are 837049. The total parameter is 3,560,889 and all parameters are trained through the model. The whole model result for the Czech language is given below.

English Vocabulary Size: 3257

English Max Length: 8

Czech Vocabulary Size: 6536

Czech Max Length: 9

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 9, 256)	1673216
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 8, 256)	0
lstm_1 (LSTM)	(None, 8, 256)	525312
time_distributed (TimeDistri	(None, 8, 3257)	837049

Total params: 3,560,889

Trainable params: 3,560,889

Non-trainable params: 0

In table 5.7 have shown the result of the training. Also, the result of the BLEU score which worked on the Training datasets in table 5.8. The comparison between training

loss and validation has shown in Figure 5.4. At first, the loss and val_loss are almost the same but in the end, val_loss is greater than loss.

Table 5.7: Source, Target and Predicted Sentences for Czech Languages (Train)

Source	Target	Predicted
prelozili text	they translated the text	they translated the text
ty knihy zde jsou moje	the books here are mine	the books here are mine
kdo te naucil psat	release the dogs	release the dogs
nas pes kousne zridka	our dog seldom bites	our dog seldom bites
nech si ten papir	keep the paper	keep the paper
hrajete squash	do you play squash	do you play squash
privedl jsem pritele	i brought a friend	i brought a friend

Table 5.8: BLEU Score for Czech Languages (Train)

BLEU-1	0.993435
BLEU-2	0.990843
BLEU-3	0.985334
BLEU-4	0.945630

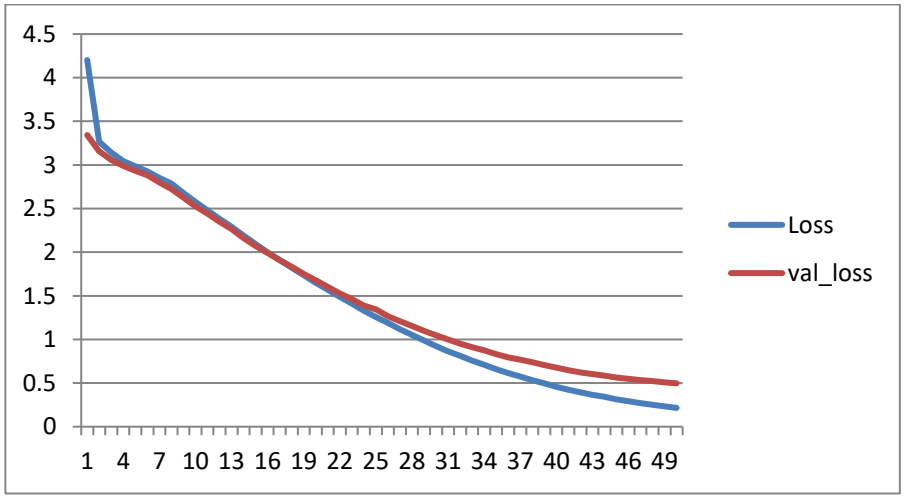


Figure 5.4: Comparison of loss and val_loss for Czech language

5.2.5 Arabic Language

Arabic is a Semitic language, like Hebrew and Aramaic. Around 292 million people speak it as their first language. Many more people can also understand it as a second language. The Arabic language has its alphabet written from right to left, like Hebrew. Arabic is another language with a non-Latin alphabet. Its 28 script letters are easier for English speakers to comprehend than the thousands of Chinese characters.For the Arabic language, the vocabulary size is 10274 and the maximum length of the vocabulary is 14. On the other hand for output as the English language, the English vocabulary size is 3523 and the maximum length of the vocabulary is 10. After going through the first embedding layer the numbers of the parameters are 2630144. All the time after passing through the two LSTM the results are 525312 parameters. The variations that happened in the parameters are in the embedding layer and after going through the time distributed layer. Here for the Arabic language in the time distributed layer 905411 parameters have detected. The total parameter is 2,319,289 and all parameters are trained through the model.The total summary of Arabic language is given below.

English Vocabulary Size: 3523
English Max Length: 10
Arabic Vocabulary Size: 10274
Arabic Max Length: 14
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 14, 256)	2630144
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 10, 256)	0
lstm_1 (LSTM)	(None, 10, 256)	525312

time_distributed (TimeDistri (None, 10, 3523) 905411

==========

Total params: 4,586,179

Trainable params: 4,586,179

Non-trainable params: 0

In table 5.9 has shown the comparison between target and predicted sentence by the model. The BLEU score has also given in Table 5.10.

Table 5.9: Source, Target and Predicted Sentence for Arabic Language (Train)

Source	Target	Predicted
هل يمكنني أن اساعدك؟	May I help you?	may i help you
أنا لا أحب السباحة في المياه المالحة	I don't like swimming in salt water.	i don't like swimming in salt water
فوزه جعله بطلاً.	His victory made him a hero.	His victory made him a hero.
الولد لطيف.	The boy is nice.	the boy is nice
علينا أن نتوقع الأسوأ.	we have to expect the worst	we have to expect the worst
اشتريت ساعة.	That's why we're here.	that's why we're here
نظف المرايا.	Clean the mirror.	clean the mirror.

Table 5.10: BLEU Score for Arabic Languages (Train)

BLEU-1	0.765843
BLEU-2	0.727017
BLEU-3	0.707661
BLEU-4	0.621634

Figure 5.5 has depicted the comparison of loss and val_loss for Arabic language.

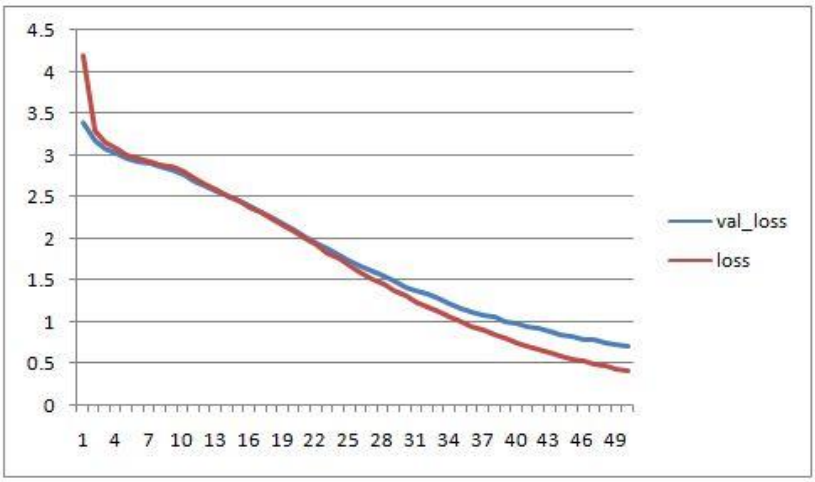


Figure 5.5: Comparison of loss and val_loss for Arabic language

5.2.6 Thai Language

Thai is unrelated to either Japanese or Korean. Thai is a tonal language and the pronunciation is very difficult. Thai is written in the Thai script, an abugida written from left to right. Many scholars believe that it is derived from the Khmer script. Here, in our research, the BLEU score for translating Thai to English is not high due to poor datasets. Although, some sentence translation is better than the Google translator. For the Thai language, the vocabulary size is 2971 and the maximum length of the vocabulary is 6. On the other hand for output as the English language, the English vocabulary size is 1977 and the maximum length of the vocabulary is 20. For the first embedding layer, 760576 parameters have found. After the first LSTM layer, the numbers of the parameters are 525312. By using two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the numbers of the parameters are the same. After going through the time distributed layer produced 508089 parameters. The total parameter is 2,319,289 and all parameters are trained through the model. The whole summary of Thai language used by this system is given here.

English Vocabulary Size: 1977

English Max Length: 20

Thai Vocabulary Size: 2971

Thai Max Length: 6

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 6, 256)	760576
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 20, 256)	0
lstm_1 (LSTM)	(None, 20, 256)	525312
time_distributed (TimeDistri	(None, 20, 1977)	508089

Total params: 2,319,289

Trainable params: 2,319,289

Non-trainable params: 0

Table 5.11: Source, Target and Predicted Sentence for Thai Language (Train)

Source	Target	Predicted
บ้านของเขาอยู่ใกล้รถไฟใต้ดิน	His house is near the subway.	his house is near the subway.
นึ่งศตวรรษคือหนึ่งร้อยปี	A century is one hundred years.	a century is one hundred years
กรุณากรอกแบบฟอร์มใบสมัครนี้	Please fill in this application form.	please fill in this application form
ไม่มีน้ำ	There is no water.	there is no water
เธอขายดอกไม้	She sells flowers.	she sells flowers
คุณจะกลับบ้านเมื่อไหร่?	When will you come home?	when will you come home
ฉันจะหาแท็กซี่ได้ที่ไหน?	Where can I get a taxi?	where can I get a taxi

In table 5.11 have shown the source, target and predicted sentence for the training datasets. The BLEU score also has shown here. There have used datasets that have contained 2910 language pairs. The more rich datasets will give better result than now.

The comparison between the loss and val_loss has shown in figure 5.6. For the Thai language in figure 5.6, the value of loss and val_loss is almost the same.

Table 5.12: BLEU Score for Thai language (Train)

BLEU-1	0.542163
BLEU-2	0.485939
BLEU-3	0.463690
BLEU-4	0.365729

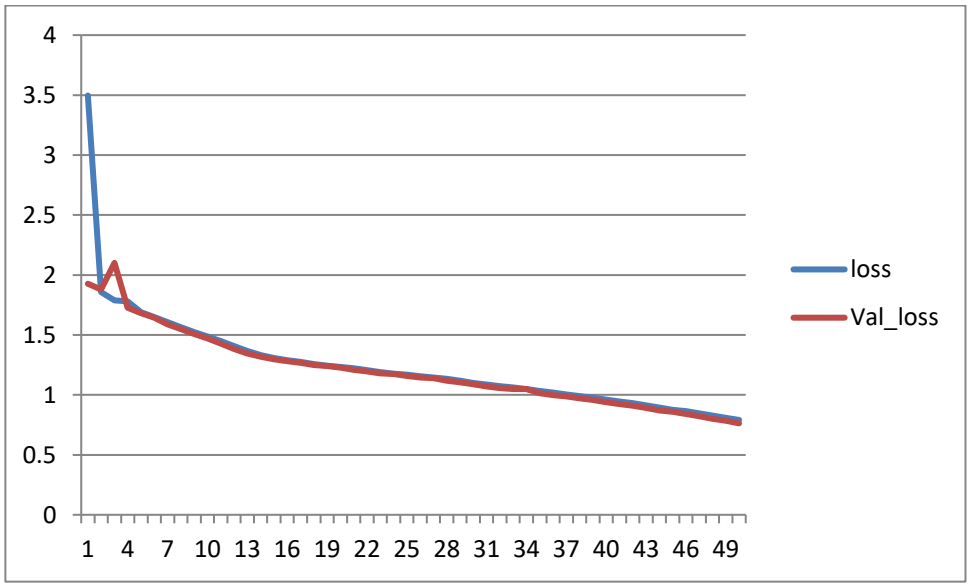


Figure 5.6: Comparison of loss and val_loss for Thai language

5.2.7 Marathi Language

Marathi is much older than Hindi or any other Indo-Aryan language. Marathi is an Indo-Aryan language spoken largely by around 83 million Marathi people of Maharashtra, India.Devanagari and Modi script are used for the Marathi language. For

Marathi, the vocabulary size is 6319 and the maximum length of the vocabulary is 9. On the other hand for output as the English language, the English vocabulary size is 3015 and the maximum length of the vocabulary is 7. Every time to train the model has gone through four layers. The first layer is the embedding layer the second is the LSTM layer, then a repeat vector is used and then again another LSTM layer is employed. Finally, go through the time distributed layer the result has found. Here the total parameter is 3,443,143 and all parameters are trained through the model. None of any parameters is left without untrained. The whole model summary for the Marathi language is given below.

English Vocabulary Size: 3015

English Max Length: 7

Marathi Vocabulary Size: 6319

Marathi Max Length: 9

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 9, 256)	1617664
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 7, 256)	0
lstm_1 (LSTM)	(None, 7, 256)	525312
time_distributed (TimeDistri	(None, 7, 3015)	774855

Total params: 3,443,143

Trainable params: 3,443,143

Non-trainable params: 0

In table 4.13 has shown the comparison of target and predicted sentences for training session.In Table 4.14 has shown the BLEU score for Marathi language for training datasets.

Table 5.13: Source, Target and Predicted Sentence for Marathi language (Train)

Source	Target	Predicted
तीत्यांनाऐकते	She listens to him	she listens to him
टॉममॅनेजरहोता	Tom was a manager	Tom was a manager
त्याजाताहेत	They are going	they are going
मीत्यांनाएकपुस्तकदिलं	I gave him a book	I gave him a book
त्यांनाप्रवासकरण्याचीसवयआहे	He is used to traveling	he is used to traveling
हेमाझेदस्तऐवज	Here are my papers	here are my papers

Table 5.14: BLEU Score for Marathi language (Train)

BLEU-1	0.701999
BLEU-2	0.642443
BLEU-3	0.592103
BLEU-4	0.423491

At last the comparison of loss and val_loss of the Marathi language have shown in figure 5.7.

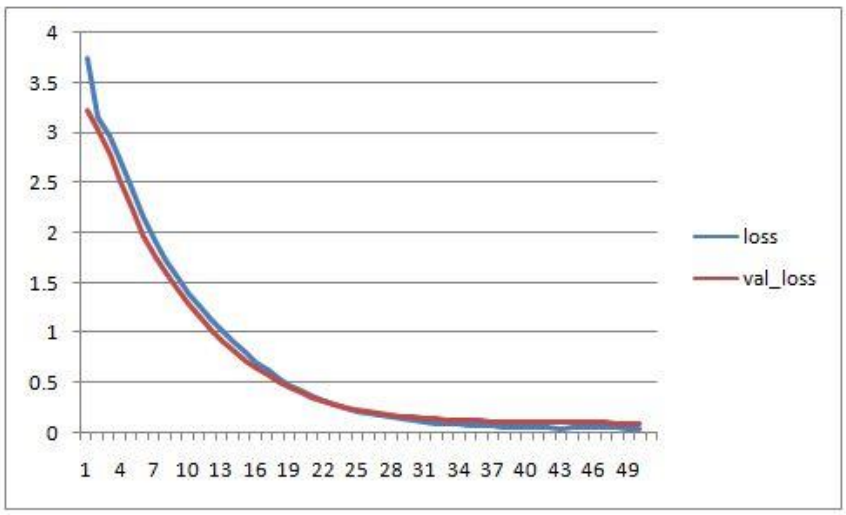


Figure 5.7: Comparison of loss and val_loss for Marathi language

5.2.8 Vietnamese Language

The Vietnamese language is the official language of Vietnam, spoken in the early 21st century by more than 70 million people. The standard language is based on the speech of educated people living in and around Hanoi. A large proportion of the vocabulary of Vietnamese has been borrowed from Chinese, and the influence of Tai languages is also evident. For the Vietnam language, the vocabulary size is 2282 and the maximum length of the vocabulary is 39. On the other hand for output as the English language, the English vocabulary size is 3500 and the maximum length of the vocabulary is 32. For the first embedding layer, the numbers of parameters are 584192. After the first LSTM layer, 525312 parameters have found. By using two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the numbers of the parameters are the same. After going through the time distributed layer 899500 parameters have observed. The total parameter is 2,534,316 and all parameters are trained through the model. The total synopsis of the model for the Vietnamese language is given below.

English Vocabulary Size: 3500
English Max Length: 32
Vietnam Vocabulary Size: 2282
Vietnam Max Length: 39
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 39, 256)	584192
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 32, 256)	0
lstm_1 (LSTM)	(None, 32, 256)	525312
time_distributed (TimeDistri	(None, 32, 3500)	899500

Total params: 2,534,316

Trainable params: 2,534,316

Non-trainable params: 0

Table 5.15 has shown the comparison between Target and predicted sentence for the Vietnamese language. Table 5.16 has shown the BLEU score. The comparison between Training loss and validation loss has shown in figure 5.8.

Table 5.15: Source, Target and Predicted Sentence for Vietnamese language (Train)

Source	**Target**	**Predicted**
Không cần giải thích	Theres no need to explain	theres no need to explain.
Cậu bé đã bắt con chim đó bằng một tấm lưới	The boy captured the bird with a net	the boy captured the bird with a net
Có lẽ tôi sẽ gọi cho cậu lúc nào đó	Maybe Ill call you sometime	maybe ill call you sometime
Túi bạn là cái nào	Which one is your bag	which one is your bag
Bạn bị ốm rồi nghỉ ngơi cho nhiều đi	Youre sick You have to rest	youre sick you have to rest
Bạn không phải là một thợ máy nhỉ	You arent a mechanic are you	youre not a mechanic are you
Tôi không muốn lấy chồng	I dont want to get marrie	i dont want to get married

Table 5.16: BLEU Score for Vietnamese language (Train)

BLEU-1	0.781726
BLEU-2	0.738286
BLEU-3	0.720109
BLEU-4	0.647246

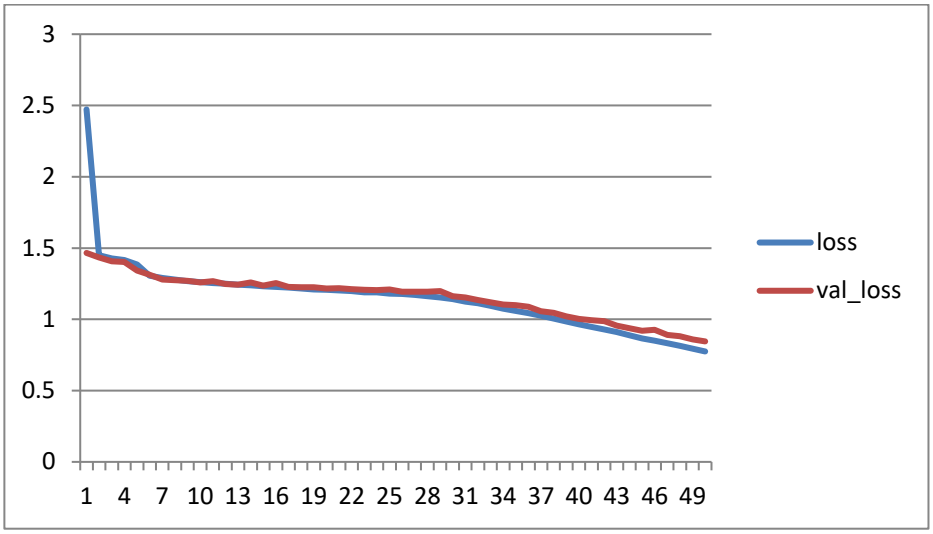


Figure 5.8: Comparison of loss and val_loss for Vietnamese language

5.2.9 Icelandic language

Icelandic is a North Germanic language spoken by about 314,000 people, the vast majority of whom live in Iceland. The language is more conservative than most other Western European languages. While most of them have greatly reduced levels of inflexion, Icelandic retains a four grammar. Since the written language has not changed much, Icelanders can read classic Old Norse literature created in the 10th through 13th centuries with relative ease. For the Icelandic language, the vocabulary size is 6233 and the maximum length of the vocabulary is 35. On the other hand for output as the English language, the English vocabulary size is 3624 and the maximum length of the vocabulary is 38. For the first embedding layer, 1595648 parameters have found. After the first LSTM layer, the numbers of the parameters are 525312. Two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the number of the parameters are not changed. After going through the time distributed layer 931368 parameters have gotten. The total parameter is 3,577,640 and all parameters are trained through the model. The Total outline for the Icelandic language is given below.

English Vocabulary Size: 3624

English Max Length: 38

Icelandic Vocabulary Size: 6233

Icelandic Max Length: 35

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 35, 256)	1595648
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 38, 256)	0
lstm_1 (LSTM)	(None, 38, 256)	525312
time_distributed (TimeDistri	(None, 38, 3624)	931368

Total params: 3,577,640

Trainable params: 3,577,640

Non-trainable params: 0

Table 5.17: Source, Target and Predicted Sentences for Icelandic Language (Train)

Source	Target	Predicted
Jörðin er hnöttótt	The earth is round	the earth is round
Hann vissi ekki hvað hann ætti að gera næst	He didnt know what to do next	he didnt know what to do next
Hver ykkar getur gert það	Any of you can do it	any of you can do it
Ég kem heim eins fljótt og ég get	Ill come back home as soon as I can	ill come back home as soon as i can
Þú munt læra hvernig á að gera þetta fyrr eða síðar。	Youll learn how to do it sooner or later	youll learn how to do it sooner or later
Hann málaði dyrnar bláar	He painted the door blue	he painted the door blue
Brandarinn hans var frábær	His joke was great	his joke was great

Table 5.18: BLEU Score for Icelandic language (Train)

BLEU-1	0.804633
BLEU-2	0.775299
BLEU-3	0.764762
BLEU-4	0.704499

In table 5.17 have shown the target and predicted sentences for the Icelandic language. The BLEU score according to the training datasets in Table 5.18.The Icelandic language is not common like Spanish, German or another language. But MT has improved a lot than before. In figure 5.9 have shown the comparison between the loss and val_loss for the Icelandic language.

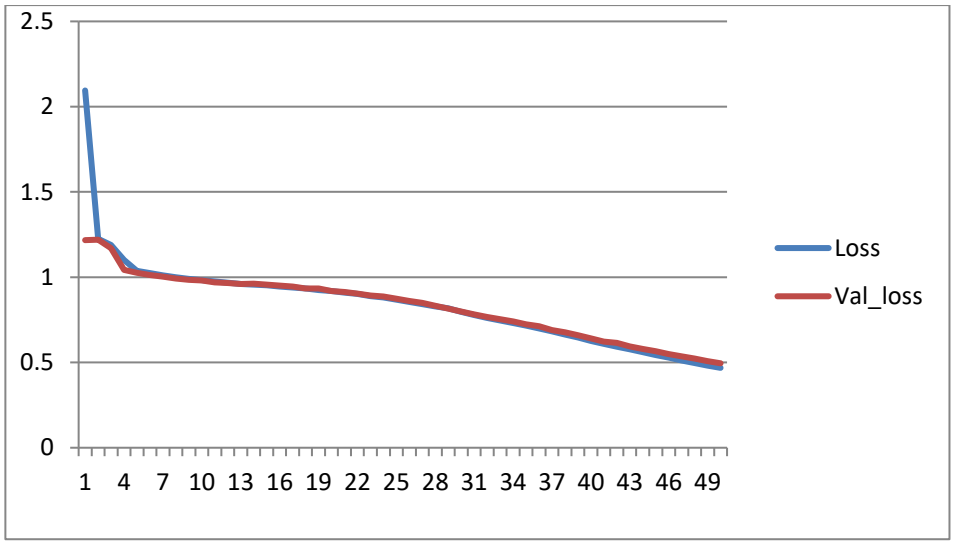


Figure 5.9: Comparison of loss and val_loss for Icelandic language

5.2.10 Macedonian Language

Macedonian is an Eastern South Slavic language which is spoken as a first language by around two million people. It serves as the official language of North Macedonia. Macedonian is also a recognized minority language in parts of Albania,

Bosnia and Herzegovina, Romania, and Serbia. It is spoken by emigrant communities mostly in Australia, Canada and the United States. For the Macedonian language, the vocabulary size is 9102 and the maximum length of the vocabulary is 10. On the other hand for output as the English language, the English vocabulary size is 4326 and the maximum length of the vocabulary is 6. For the first embedding layer, we have got 2330112 parameters. After the first LSTM layer, we have got 525312 parameters. By using two LSTM layers and a repeat vector for converting the matrix from 2D to 3D the numbers of the parameters are the same. After going through the time distributed layer we have got 1111782 parameters. The total parameter is 4,492,518 and all parameters are trained through the model. The total sketch of the model for the Macedonian language is given below.

.English Vocabulary Size: 4326

English Max Length: 6

Macedonian Vocabulary Size: 9102

Macedonian Max Length: 10

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 10, 256)	2330112
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 6, 256)	0
lstm_1 (LSTM)	(None, 6, 256)	525312
time_distributed (TimeDistri	(None, 6, 4326)	1111782

Total params: 4,492,518

Trainable params: 4,492,518

Non-trainable params: 0

In Table 5.19 the target and predicted sentence for the Macedonian language is given below. Table 5.20 has shown the BLEU score for the training dataset.

Table 5.19: Source, Target and Predicted Sentences for Macedonian Language (Train)

Source	Target	Predicted
Том живее во близина	Tom lives close by	tom lives close by
Тука ми е поарно	Im better off here	im better off here
Побогат сум од тебе	Im richer than you	im richer than you
Том е влијателен	Tom is influential	tom is influential
Колку си беден	Youre so pathetic	youre so pathetic
Дома сум	Im at home	im home
Спушти го тој пиштол	Put down that gun	put down that gun

Table 5.20: BLEU Score for Macedonian language (Train)

BLEU-1	0.674925
BLEU-2	0.605221
BLEU-3	0.537066
BLEU-4	0.323117

In figure 5.10 have shown the comparison between training loss and validation loss for Macedonian language.

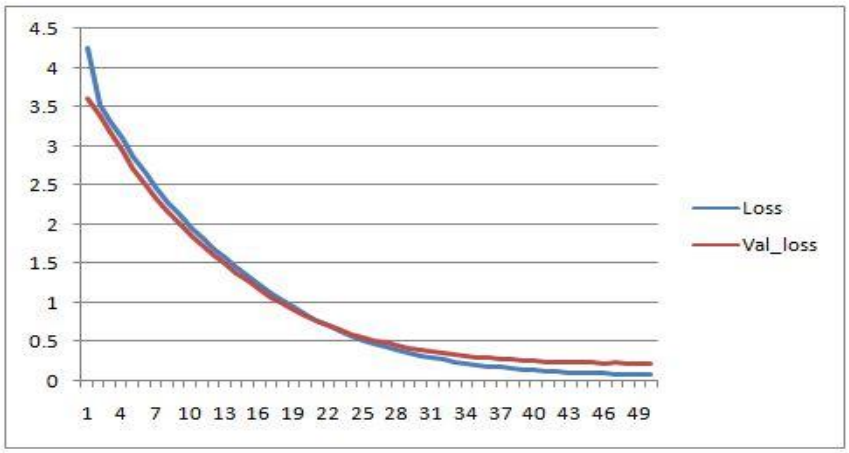


Figure 5.10: Comparison of loss and val_loss for Vietnamese language

5.2.11 Kablye language

Kabyle is the Berber language spoken by the Kabyle people in the north and northeast of Algeria. Kabyle is spoken primarily in Kabylia, east of the capital Algiers and in Algiers itself. For Kabyle, the vocabulary size is 6862 and the maximum length of the vocabulary is 10. On the other hand for output as the English language, the English vocabulary size is 1898 and the maximum length of the vocabulary is 6. Every time to train the model has gone through four layers. The first layer is the embedding layer the second is the LSTM layer, then a repeat vector and another LSTM layer is used. At last, the time distributed layer is worked. Here the total parameter is 3,295,082 and all parameters are trained through the model. None of any parameters is left without untrained.256 size embedding layer has used here. The total outline of the Kabyle language for this model is given below.

English Vocabulary Size: 1898

English Max Length: 6

Kabyle Vocabulary Size: 6862

Kabyle Max Length: 10

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 10, 256)	1756672
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 6, 256)	0
lstm_1 (LSTM)	(None, 6, 256)	525312
time_distributed (TimeDistri	(None, 6, 1898)	487786

Total params: 3,295,082

Trainable params: 3,295,082

Non-trainable params: 0

Table 5.21: Source, Target and Predicted Sentences for Kabyle Language (Train)

Source	Target	Predicted
Tibḥirtiw d tamecṭuḥt	My garden is small	my garden is small
Ṛeqqeɛ waki	Fix this	fix this
Ttuɣ kullec ɣef waya	I forgot all about that	i forgot all about that
Tameṭṭut n Tom d mmtismin	Tom has a jealous wife	tom has a jealous wife
Ad mdyecnu Tom	Tom will sing for you	tom will sing for you
D acu i nezmer ad kenttid nexdem	What can we do for you	what can we do for you
Ur ttɛaqabet ara Tom ɣef ayen	Dont punish Tom for that	dont punish tom for that

Table 5.22: BLEU Score for Kabyle language (Train)

BLEU-1	0.686006
BLEU-2	0.623655
BLEU-3	0.577418
BLEU-4	0.438545

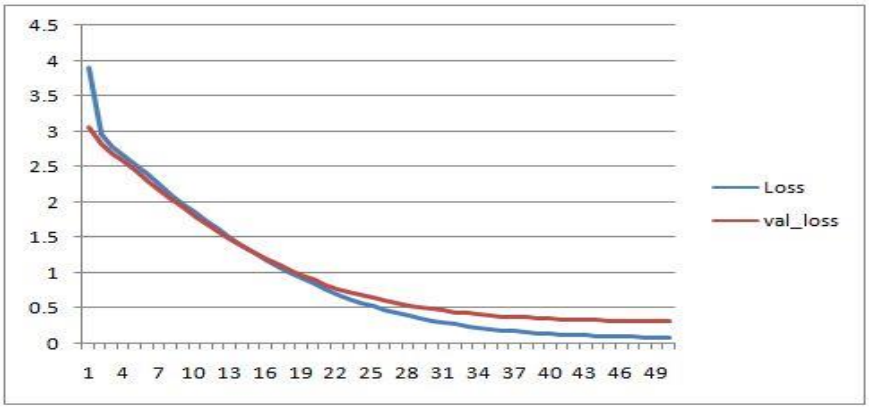


Figure 5.11: Comparison of loss and val_loss for Kablye language

In Table 5.21 have shown the target and predicted sentence for the Kabyle language. The BLEU score for the training session has also shown in Table 5.22. In figure 5.11 the comparison between loss and val_loss has depicted through the curve. There have shown the first 50 epoch of loss and val_loss for this language.Kablye language is a uncommon language which scripts is little similar with English.

5.2.12 Tagalog Language

Tagalog is an Austronesian language spoken as a first language by the ethnic, Tagalog people. This language vocabulary has been much influenced by Spanish and English, to some extent by Chinese, Sanskrit, Tamil, and Malay. Tagalog is the primary language of Manila. Tagalog is currently written using the Latin alphabet.Many people even wonder Filipino and Tagalog are the same languages. But they are not. Instead, Filipino language is the developing from of Tagalog. For the Tagalog language, the vocabulary size is 3229 and the maximum length of the vocabulary is 33. On the other hand for output as the English language, the English vocabulary size is 2538 and the maximum length of the vocabulary is 32. Every time to train the model has gone through different layers. The first layer is the embedding layer the second is the LSTM layer, a repeat vector and another LSTM layer is also used. And the last layer is time distributed layer.Here the total parameter is 2,529,514 and all parameters are trained through the model. None of any parameters is left without untrained. The total summary for the Tagalog language of this model is given below.

English Vocabulary Size: 2538
English Max Length: 32
Tagalog Vocabulary Size: 3229
Tagalog Max Length: 33
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 33, 256)	826624
lstm (LSTM)	(None, 256)	525312

repeat_vector (RepeatVector) (None, 32, 256) 0

lstm_1 (LSTM) (None, 32, 256) 525312

time_distributed (TimeDistri (None, 32, 2538) 652266

Total params: 2,529,514

Trainable params: 2,529,514

Non-trainable params: 0

Table 5.23: Source, Target and Predicted Sentences for Tagalog Languages (Train)

Source	Target	Predicted
Tinanong ako ng isang dayuhan kung nasaan ang estasyon	A foreigner asked me where the station was	my foreigner asked me where the station was
Bakit di mo sabihin sa kanya nang diretso	Why dont you tell her directly	why dont you tell her directly
Kumanta siya ng kanta	He sang a song]	he sang a song
Matangkad siya at guwapo	He is tall and handsome	he is tall and handsome
Nakauwi na ako	Im home	im home
Pinuri siya ng kanyang guro	Her teacher praised her	her teacher praised her
Biro ito tama	This is a joke right	this is a joke right

Table 5.24: BLEU Score for Tagalog languages (Train)

BLEU-1	0.769881
BLEU-2	0.729119
BLEU-3	0.709919
BLEU-4	0.631550

In table 5.23 have shown the comparison between target and predicted sentences. The BLEU score is also given in Table 5.24. In figure 5.24 has shown the graph of loss and val_loss of the model for the Tagalog language.

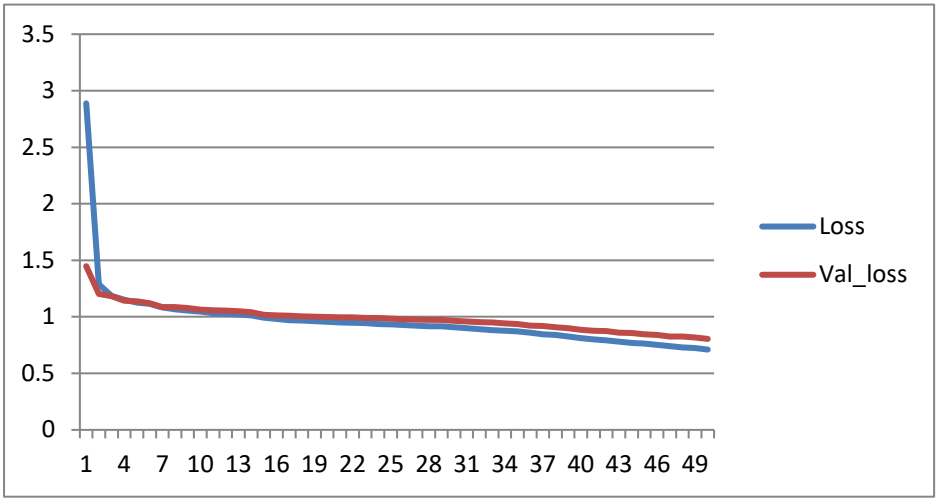


Figure 5.12: Comparison of loss and val_loss for Tagalog language

5.2.13 Cantonese language

Cantonese is a language within the Chinese branch of the Sino-Tibetan languages. The language is the traditional prestige variety of the Yue Chinese dialect group. Cantonese is spoken largely in Hong Kong, as well as in Macau and the Guangdong province, including Guangzhou. For Cantonese, the vocabulary size is 3208 and the maximum length of the vocabulary is 4. On the other hand for output as the English language, the English vocabulary size is 2671 and the maximum length of the vocabulary is 32. Every time to train the model has gone through different layers. The first layer is the embedding layer the second is the LSTM layer. A repeat vector is used and again another LSTM layer is applied. And at last, we go through the time distributed layer. After we go through these four layers finally we got the result. Here the total parameter is 2,558,319 and all parameters are trained through the model. None of any parameters are left without untrained. For the embedding layer, we have used 256 sizes embedding layer. The embedding that there have used is 256. The summary of the model of the language is given below.

English Vocabulary Size: 2671

English Max Length: 32

Cantonese Vocabulary Size: 3208

Cantonese Max Length: 4

Model: "sequential"

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 4, 256)            821248
_________________________________________________________________
lstm (LSTM)                  (None, 256)               525312
_________________________________________________________________
repeat_vector (RepeatVector) (None, 32, 256)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 32, 256)           525312
_________________________________________________________________
time_distributed (TimeDistri (None, 32, 2671)          686447
=================================================================
```

Total params: 2,558,319

Trainable params: 2,558,319

Non-trainable params: 0

In table 5.25 have shown a comparison between target and predicted sentence base on the source sentence. In table 5.26 BLEU score has shown for the language

Table 5.25: Source, Target and Predicted Sentences for Cantonese Language (Train)

Source	Target	Predicted
我冇錢，但係我有夢想。	I have no money but I have dreams	i have no money but i have dreams
佢仲要問佢阿爸阿媽攞錢。	She is still financially dependent on her parents	she is still financially dependent on her parents
你會唔會覺得我份人太過物質呀？	Do you think Im too materialistic	do you think im too materialistic
我想學跳舞。	I would like to learn how to dance	i would like to learn how to dance
試吓從我個角度去諗吖。	Try to look at it from my point of view	try to look at it from my point of view
你咁嘅成績入唔到大學㗎喎。	With those results you wont be able to go to university	with those results you wont be able to go to university
我等咗我個friend成粒鐘。	Ive spent an entire hour waiting for my friend	ive spent an entire hour waiting for my friend

Table 5.26: BLEU Score for Cantonese languages (Train)

BLEU-1	0.818451
BLEU-2	0.787867
BLEU-3	0.776278
BLEU-4	0.718275

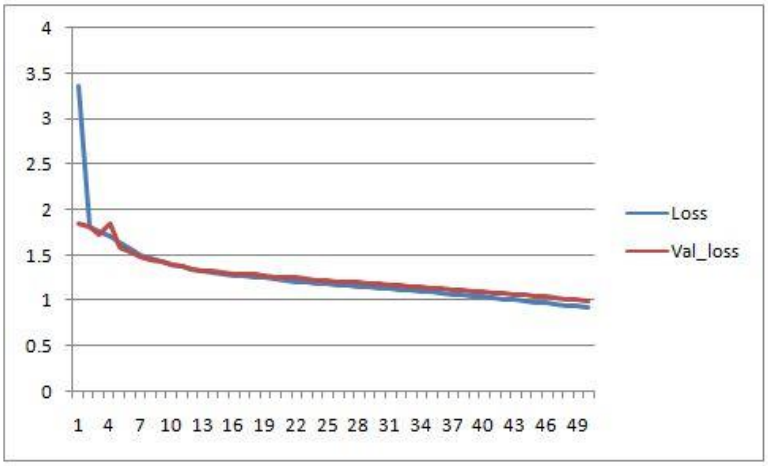


Figure 5.13: Comparison of loss and val_loss for Cantonese language

.

Figure 5.14 has shown a curve by comparing the loss and val_loss for the Cantonese language.

5.2.14 Low German language

Low German or Low Saxon is a West Germanic language variety spoken mainly in Northern Germany and the northeastern part of the Netherlands. The language is also spoken to a lesser extent in German emigration worldwide. Before going through the training the established model has worked with vocabulary size and maximum length. The neural translation model is a sequential model. All the work of the model has done sequentially. For low Saxon, the vocabulary size is 2725 and the maximum length of the vocabulary is 40. On the other hand for output in English, the English vocabulary size is

2539 and the maximum length of the vocabulary is 2539.The first layer is the embedding layer the second is the LSTM layer, then a repeat vector is used and then we another LSTM layer is utilized. The last layer is the time distributed layer. Here the total parameter is 2,400,747 and all parameters are trained through the model. The embedding size defines the length of the vectors used to represent words. A larger dimensionality will result in a more expressive representation, and in turn, better skill. Interestingly, the results show that the largest size tested did achieve the best results, but the benefit of increasing the size was minor overall. For the embedding layer 256 sizes embedding layer is used. The total summary of low German language is given below.

English Vocabulary Size: 2539
English Max Length: 38
Low Saxon Vocabulary Size: 2725
Low Saxon Max Length: 40
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 40, 256)	697600
lstm (LSTM)	(None, 256)	525312
repeat_vector (RepeatVector)	(None, 38, 256)	0
lstm_1 (LSTM)	(None, 38, 256)	525312
time_distributed (TimeDistri	(None, 38, 2539)	652523

Total params: 2,400,747
Trainable params: 2,400,747
Non-trainable params: 0

In table 5.27 have shown a comparison between target and predicted sentence base on the source sentence. In table 5.28 BLEU score has shown for the language.

Table 5.27: Source, Target and Predicted Sentences for Low German Languages(Train)

Source	Target	Predicted
He hett twee Böker utlehnt	He borrowed two books	he borrowed two books
Washington sien Armee hett Trenton innahmen	Washingtons army has captured Trenton	washingtons army has captured Trenton
Ik bün na 'n Markt gahn	I went to the market	i went to the market
De Ammel weer vull mit Water	The bucket was full of water	the bucket was full of water
Ik bün gornich mood	Im not at all tired	im not at all tired
Do dien Mütz op	Put on your cap	put on your cap
Indien is 1947 von Grootbritannien unafhängig worn	India gained independence from Britain in 1947	india gained independence from britain in 1947

Table 5.28: BLEU Score for Low German languages (Train)

BLEU-1	0.785786
BLEU-2	0.753492
BLEU-3	0.740175
BLEU-4	0.668968

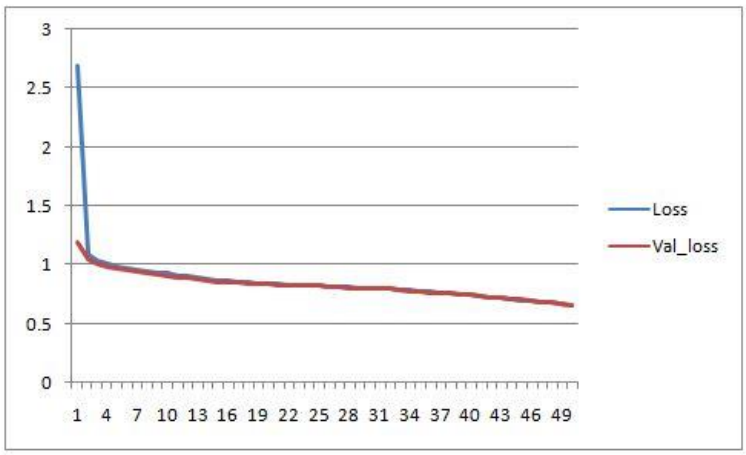


Figure 5.14: comparison of loss and val_loss for Low German language

figure 5.14 have shown the comparison between the training loss and validation loss for the Low German language.

Loss is calculated based on the train set, val_loss or validation loss. The training loss is the average of the losses over each batch of training data. Because any model is changing over time, the loss over the first batches of an epoch is generally higher than over the last batches. On the other hand, the testing loss for an epoch is computed using the model as it is at the end of the epoch, resulting in a lower loss. That's not correct and validation loss at the end of the epoch is computed using a separate set of data that the model never saw. Validation loss can be lower or higher or equal to the training loss. No matter whether a loss or val_loss, both is a loss function and both loss and val_loss should be decreased. There are times that loss is decreasing while validation loss is increasing which means learning based on training data is done successfully but due to many factors this learning isn't that understanding to be able to be applied on another data-set that hasn't been seen(validation data set). Accordingly, experts believe that the model is overlearned. The training loss generally is lower than the valid loss because the validation dataset is used to validate the model with data that the model has never seen. For this reason, the valid loss generally is higher compared with the training loss. If we are lucky the values could be similar. In machine learning and deep learning, there are three cases. Underfitting happens where loss > validation loss, but only slightly if the loss is far higher than validation loss. Overfitting happens where loss < validation loss This means that model is fitting very nicely the training data but not at all the validation data, in other words, it's not generalizing correctly to unseen data.If losses == validation loss then is called Perfect fitting. If both values end up being roughly the same and also if the values are converging (plot the loss over time) then chances are very high that the model works perfectly. Here for every language, the plots of validation loss and training loss have drawed. According to the result, that both val_loss and loss is decreasing along with epoch.Although some underfitting problem, overfitting problem is identified but all in all the model works well. Val_loss and epochs make a great impact on translation accuracy. If val_loss did not improve so as the translation accuracy does not improve much. On other higher datasets ensure correct Translation. Whenever the more need time to train the model the better the model is trained. According to the training and testing results, the more the epoch the more reduced the val_loss. After 100 Epoch,

val_loss reduced from 1.8101 to 0.0674, for the Bengali Language. After 100 epochs, the reduction is from 3.4536 to 0.4044, for the Romanian Language. After 100 epoch, for the Arabic Language val_loss reduced from 3.3947 to 0.4532. After 100 epochs, the reduction is from 3.2309 to 0.1071, for the Marathi Language. After 100 epoch, val_loss reduced from 1.9258 to 0.1150, for the Thai Language. Here have shown the plot of these 14 languages of val_loss and loss for 50 epochs. The decreasing of val_loss of the first 50 epochs for every language is shown through the graph. By increasing the word embedding layer the val_loss decreased surprisingly but does affect the translation quality. Here, for all languages 256 embedding layer is used. The reason behind using 14 languages datasets for the experiment is every language gives a new conclusion about a different perspective of the model. Val_loss has a big impact on translation accuracy. The more the val_loss improved the higher we got the BLEU score. If the val_loss did not decrease then the translation output also goes worse. The BLEU score is also given here for every language. For every language translate from a different language to English 14 sentences have shown in the training and testing session to validate the translation quality of the model. But the research has worked with different datasets which have contained sentences pairs from 2000 to 30000.

5.3 TESTING RESULTS AND ANALYSIS

First of all the training result and the BLEU score of every language have analysed. In this section the testing results have shown for every languages.The BLEU score of every language for testing result has given below.

5.3.1 Bengali language

In the previous section the training result, how model worked with every language, loss function and BLEU score according to training datasets have discussed. In this section the testing result of every languages and BLEU score has shown.The comparison between Target and predicted sentence for Bengali language have considered in Table 5.29. The model can perfectly predict the target sentence. In table 5.30 have given the BLEU score of testing datasets. The BLEU score shows that the

translation quality is very high. Also in figure 5.15 the screenshot of training and testing result that found for Bengali languages have shown.

Table 5.29: Source, Target and Predicted Sentences for Bengali language (Test)

Source	Target	Predicted
আপনি কে	who are you	who are you
ওরা পরোয়া করে না।	im coming back	im coming back
তোমার কাছে একটা পেন হবে|	do you have a pen on you	do you have a pen on you
আমার কাছে মাত্র একটাই কম্বল আছে।	i only have one blanket	i only have one blanket
আমি একটা অ্যাকাউন্ট খুলতে চাই।	id like to open an account	id like to open an account
আমি আপনার উপর নির্ভর করে আছি।	i am counting on you	i am counting on you

Table 5.30: BLEU Score for Bengali Language (Test)

BLEU-1	0.948236
BLEU-2	0.924976
BLEU-3	0.895370
BLEU-4	0.825149

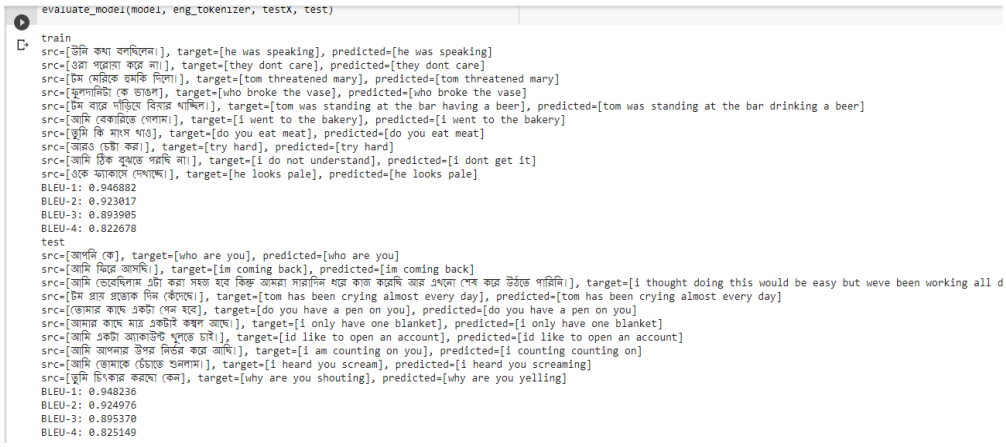


Figure 5.15: Train and Testing Result for Bengali language

5.3.2 Romanian Language

Table 5.31: Source, Target and Predicted sentence for Romanian Language (Test)

Source	Target	Predicted
a facut totul din bunatate	he did it all out of kindness	he did it all out of kindness
pulsul meu e slab	my pulse is weak	my pulse is weak
am fost 91resent ieri la scoala	I was present at school yesterday	I was present at school yesterday
uitate sub pat	look under the bed	look under the bed
am doua pisici	I have two cats	I have two cats
biciclistul a urcat dealul	the cyclist climbed the hill	the cyclist climbed the hill
trebuie sa ne prioritizam nevoile	we have to prioritize our needs	we have to prioritize our needs

In table 5.31 have shown the testing result for Romanian language. The target and predicted sentence is same almost same. The BLEU score has shown in table 5.32.The final BLEU score is 0.862303.That represents that the model has given high quality translation for Romanian language for translating into English. Figure 5.16 have shown the overall result of training and testing datasets and also the BLEU score.

Table 5.32: BLEU Score for Romanian Languages (Test)

BLEU-1	0.917704
BLEU-2	0.899233
BLEU-3	0.895238
BLEU-4	0.862303

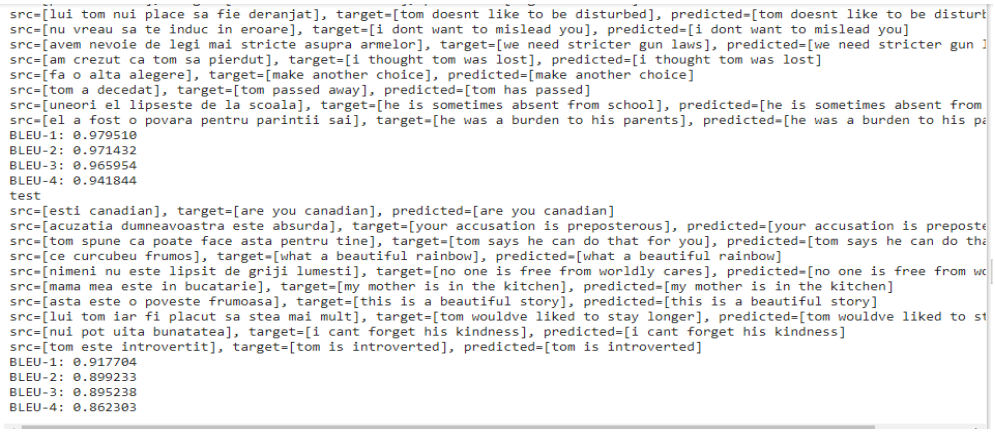


Figure 5.16: Train and Testing Result for Romanian language

5.3.3 Danish Language

The testing result for the Danish language and BLEU score has shown in Table 5.39 and Table 5.40. In every language, the testing result has shown according to the sentence by sentence. First, there has a source sentence and target sentence. Then according to the source sentence, the model has predicted the predicted sentence.In the first table, the comparison has exhibited. In the second table the BLEU scores that have shown. The final BLEU score is 0.821216.

Table 5.33: Source, Target and Predicted Sentences for Danish Language (Test)

Source	Target	Predicted
du kan ikke sige nej	you cant say no	you cant say no
tom vil maske vidne	tom might testify	tom might testify
thomas er fattig	tom is poor	tom is poor
jeg er en smule sulten	im a bit hungry	im a bit hungry
jeg har mistet min pung	ive lost my wallet	ive lost my wallet
hun var nedtrykt	she felt blue	she felt blue
kan jeg hjlpe dig pa nogen made	can i help you in any way	can i help you in any way

Table 5.34: BLEU Score for Danish Language (Test)

BLEU-1	0.918473
BLEU-2	0.895433
BLEU-3	0.884174
BLEU-4	0.821216

Figure 5.17 have shown the result of training and testing datasets and also the BLEU score.

```
src=[jeg er bedre], target=[i am better], predicted=[im better better]
src=[jeg tror tom kan lide mary], target=[i think tom likes mary], predicted=[i think tom likes mary]
src=[jeg vil fortlle dig sandheden], target=[ill tell you the truth], predicted=[ill tell you the truth]
src=[tom arbejder], target=[tom is working], predicted=[tom is]
src=[aftensmaden er klar], target=[dinner is ready to eat], predicted=[dinners ready]
src=[noget ma der gres], target=[something must be done], predicted=[something must be done]
BLEU-1: 0.965638
BLEU-2: 0.953126
BLEU-3: 0.942366
BLEU-4: 0.886245
test
src=[du kan ikke sige nej], target=[you cant say no], predicted=[you cant say no]
src=[tom vil maske vidne], target=[tom might testify], predicted=[tom might testify]
src=[thomas er fattig], target=[tom is poor], predicted=[tom is poor]
src=[jeg er en smule sulten], target=[im a bit hungry], predicted=[im a bit hungry]
src=[jeg har mistet min pung], target=[ive lost my wallet], predicted=[ive lost my wallet]
src=[hun var nedtrykt], target=[she felt blue], predicted=[she felt blue]
src=[kan jeg hjlpe dig pa nogen made], target=[can i help you in any way], predicted=[can i help you in any way]
src=[regner det nu], target=[is it raining now], predicted=[is it raining now]
src=[tom er endnu ikke teenager], target=[tom isnt a teenager yet], predicted=[tom isnt a teenager yet]
src=[ls denne bog], target=[read this book], predicted=[read this book]
BLEU-1: 0.918473
BLEU-2: 0.895433
BLEU-3: 0.884174
BLEU-4: 0.821216
```

Fig 5.17: Train and Testing Result for Danish language

5.3.4 Czech Language

In table 5.41 the testing result for the Czech language has shown. The target and predicted sentences have featured here. In table 5.42 has shown the BLEU score. The overall result has shown in the screenshot in figure 5.21 below.

Table 5.35: Source, Target and Predicted Sentences for Czech Language (Test)

Source	Target	Predicted
tom se nevzdal	tom didnt give up	tom didnt give up
zdavalo se mi o tobe	i used to dream about you	i used to dream about you
Nespi	he doesnt sleep	he doesnt sleep
tom se rychle osprchoval	tom took a quick shower	tom took a quick shower
tom mi opravil esej	tom corrected my essay	tom corrected my essay
bylo to velmi dobre	that was very good	that was very good
kde je ta brana	where is the gate	where is the gate

Table 5.36: BLEU Score for Czech Language (Test)

BLEU-1	0.932431
BLEU-2	0.918535
BLEU-3	0.914338
BLEU-4	0.866760

```
src=[ty knihy zde jsou moje], target=[the books here are mine], predicted=[the books here are mine]
src=[kdo te naucil psat], target=[who taught you to write], predicted=[who taught you to write]
src=[pustte ty psy], target=[release the dogs], predicted=[release the dogs]
src=[nas pes kousne zridka], target=[our dog seldom bites], predicted=[our dog seldom bites]
src=[dosly nam moznosti volby], target=[were out of options], predicted=[were out of options]
src=[nech si ten papir], target=[keep the paper], predicted=[keep the paper]
src=[hrajete squash], target=[do you play squash], predicted=[do you play squash]
src=[privedl jsem pritele], target=[i brought a friend], predicted=[i brought a friend]
src=[prelozili text], target=[they translated the text], predicted=[they translated the text]
src=[proc to tom udelal], target=[what made tom do it], predicted=[what made tom do it]
BLEU-1: 0.993435
BLEU-2: 0.990843
BLEU-3: 0.985334
BLEU-4: 0.945630
test
src=[tom se nevzdal], target=[tom didnt give up], predicted=[tom didnt give up]
src=[zdavalo se mi o tobe], target=[i used to dream about you], predicted=[i used to dream about you]
src=[nespi], target=[he doesnt sleep], predicted=[he doesnt sleep]
src=[tom se rychle osprchoval], target=[tom took a quick shower], predicted=[tom took a quick shower]
src=[tom mi opravil esej], target=[tom corrected my essay], predicted=[tom corrected my essay]
src=[bylo to velmi dobre], target=[that was very good], predicted=[that was very good]
src=[kde je ta brana], target=[where is the gate], predicted=[where is the gate]
src=[hodne te obdivuji], target=[i admire you a lot], predicted=[i admire you a lot]
src=[potrebuju abys poslouchal], target=[i need you to listen], predicted=[i need you to listen]
src=[tom mi jeden poslal], target=[tom sent one to me], predicted=[tom sent one to me]
BLEU-1: 0.932431
BLEU-2: 0.918535
BLEU-3: 0.914338
BLEU-4: 0.866760
```

Fig 5.18: Train and Testing Result for Czech language

5.3.5 Arabic Language

Table 5.33 have shown the target and predicted sentence according to the source sentence.Table 5.34 has shown the BLEU score of the testing datasets that we have used. Finally, figure 5.17 have given the overall picture of the training and testing session.

Table 5.37: Source, Target and Predicted Sentence for Arabic Languages (Test)

Source	Target	Predicted
هل يمكنني أن اساعدك؟	May I help you?	may i help you
أنا لا أحب السباحة في المياه المالحة	I don't like swimming in salt water.	i don't like swimming in salt water
فوزه جعله بطلاً.	His victory made him a hero.	His victory made him a hero.
الولد لطيف.	The boy is nice.	the boy is nice
علينا أن نتوقع الأسوأ.	we have to expect the worst	we have to expect the worst
اشتريت ساعة.	That's why we're here.	that's why we're here
نظف المرايا.	Clean the mirror.	clean the mirror.

Table 5.38: BLEU Score for Arabic language (Test)

BLEU-1	0.708758
BLEU-2	0.666188
BLEU-3	0.650951
BLEU-4	0.565170

Arabic language has many varities with the pronounciantion and accent.Also the scripts of Arabic language is unique differ from English and other language.For every aspects translation from Arabic to English is essential.So, good quality translation is the main concern of these research.

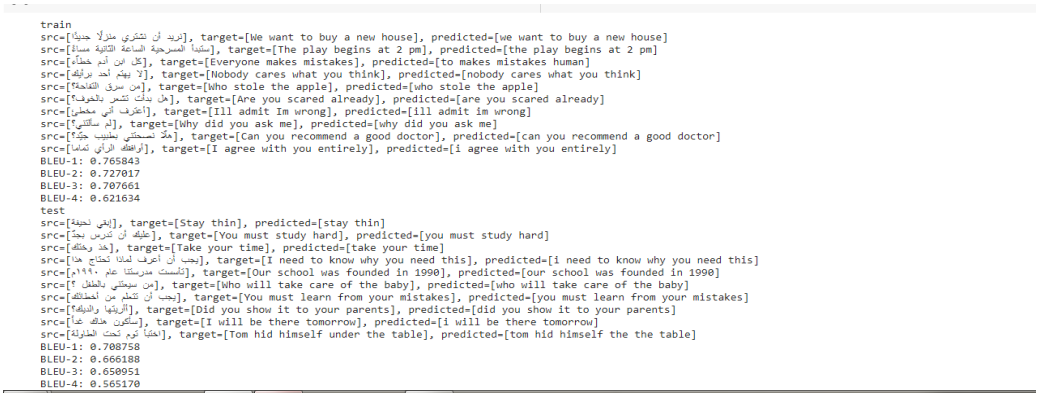


Fig 5.19: Train and Testing Result for Arabic language

5.3.6 Thai Language

Table 5.39: Source, Target and Predicted Sentence for Thai Languages (Test)

Source	Target	Predicted
ฉันรู้จักร้านอาหารอิตาเลียนที่ดี	I know a good Italian restaurant.	i know a good italian restaurant
อย่าหยุดเขา	Don't stop him.	don't stop him
เรากำลังกินแอปเปิ้ล	We're eating apples.	Were eating apples
ลองกัดชิมดูสักคำซิ	Take a bite.	take a bite
ฉันไม่จำเป็นต้องตอบคำถามของคุณ	I don't have to answer your question.	i don't have to answer your question
หนังสือกองนี้ของฉัน ส่วนกองโน้นน่ะของเค้า	These books are mine and those books are his.	these books are mine and those books are his

Table 5.39 have shown the source, target and predicted sentences for testing.

Table 5.40: BLEU Score for Thai language (Test)

BLEU-1	0.545102
BLEU-2	0.488571
BLEU-3	0.466125
BLEU-4	0.368832

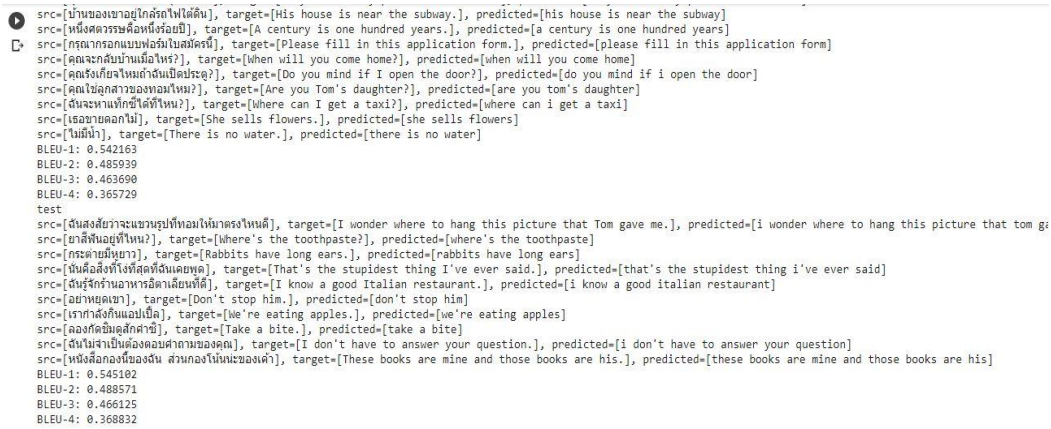


Fig 5.20: Train and Testing Result for Thai language

BLEU score has shown in table 5.40 and the overall result has exhibited in figure 5.20.

5.3.7 Marathi Language

Table 5.37 have shown the target and predicted sentence for the Marathi language of this model, the BLEU score of testing sets have given in table 5.38.

Table 5.41: Source, Target and Predicted Sentence for Marathi language (Test)

Source	Target	Predicted
मला सफरचंद खायला आवडतात	I like to eat apples	I like to eat apples
ट्रेन येतेय	The train is coming	the train is coming
मी शांत राहिलो	I remained quiet	I remained quiet
माझ्याकडे नव्हतं	I have it	II have it
टॉमला तुला भेटायचं होतं	Tom wanted to meet you	tom wanted to meet you
मी डाउनलोड केलं	I downloaded it	I downloaded it

Table 5.42: BLEU Score for Marathi language (Test)

BLEU-1	0.691874
BLEU-2	0.631145
BLEU-3	0.581604
BLEU-4	0.414880

```
src=[जुना आहे], target=[Its old], predicted=[its old]
src=[पुढच्या रविवारचा काय विचार आहे], target=[What about next Sunday], predicted=[what about next sunday]
src=[तू काय वेडा झाला आहेस का], target=[Have you gone crazy], predicted=[have you gone crazy]
src=[मला हा चहा आवडतो], target=[I like this tea], predicted=[i like this tea]
src=[तू त्याला फोन केलास का], target=[Did you phone him], predicted=[did you phone him]
src=[मला तुम्हाला परत पाहायचं आहे], target=[I want to see you again], predicted=[i want to see you again]
src=[खरोखरचं होतं], target=[It was real], predicted=[it was real]
src=[तूसुद्धा जोतेयस], target=[Are you going too], predicted=[are you going too]
src=[त्यांनी त्यांना जोरात मारलं], target=[She hit him hard], predicted=[she hit him hard]
BLEU-1: 0.701999
BLEU-2: 0.642443
BLEU-3: 0.592103
BLEU-4: 0.423491
test
src=[टॉम सोडून सगळे आले], target=[Everyone but Tom came], predicted=[everyone but tom came]
src=[तुम्हाला फळं आवडतात असं वाटतंय], target=[You seem to like fruit], predicted=[you seem to like fruit]
src=[तू जिंकला आहेस], target=[Youve won], predicted=[youve won]
src=[तू कुठे पोहतोस], target=[Where do you swim], predicted=[where do you swim]
src=[मी जिंकलेय], target=[Ive won], predicted=[ive won]
src=[तुम्ही राक्षस आहात], target=[Youre a monster], predicted=[youre a monster]
src=[तो अगदी मागेच आहे तुझ्या], target=[Hes right behind you], predicted=[hes right behind you]
src=[कोणालाही भूक लागली नाहीये], target=[Nobodys hungry], predicted=[nobodys hungry]
src=[मी कशाला हसेन], target=[Why would I laugh], predicted=[why would i laugh]
src=[टॉम गाईल], target=[Tom will sing], predicted=[tom will sing]
BLEU-1: 0.691874
BLEU-2: 0.631145
BLEU-3: 0.581604
BLEU-4: 0.414880
```

Fig 5.21: Train and Testing Result for Marathi language

5.3.8 Vietnamese Language

The target and predicted sentences for the Vietnamese language have shown below in Table 5.43.

Table 5.43: Source, Target and Predicted Sentences for Vietnamese Language (Test)

Source	Target	Predicted
Ấy có sợ chó không	Are you afraid of dogs	are you afraid of dogs
Chuyện gì đã qua thì cho qua	It is no use crying over spilt milk	there is no use crying over spilt milk
Chúng tôi đã có một trải nghiệm không mấy dễ chịu ở đó	We had an unpleasant experience there	we had an unpleasant experience there
Đi qua cái cửa màu cam	Go through the orange door	go through the orange door
tao sẽ nhớ mày	I will miss you	i will miss you
Cuốn sách này tôi mua ở hiệu sách trước nhà ga	I bought this book at the book store in front of the station	i bought this book at the book store in front of the station
Hôm nay tôi không thể làm việc	I cant work today	i cant work today

After comparing the predicted sentences with the target sentence in table 4.44 we have shown the BLEU score of the testing session.

Table 5.44: BLEU Score for Vietnamese Languages (Test)

BLEU-1	0.664341
BLEU-2	0.613657
BLEU-3	0.602967
BLEU-4	0.528193

Below in figure 5.22 have given the overall scenario of the training and testing result.

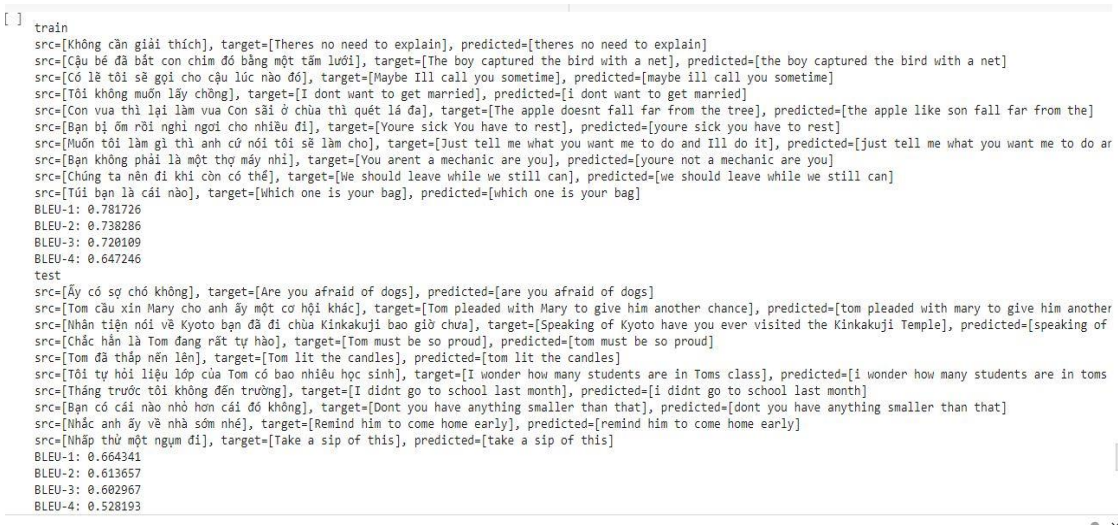


Fig 5.22: Train and Testing Result for Vietnamese language

5.3.9 Icelandic Language

In table 5.45 we have shown the comparison of target and predicted sentences for the Icelendic language and BLEU score has given in table 5.46.

Table 5.45: Source, Target and Predicted Sentences for Icelandic Languages (Test)

Source	Target	Predicted
Sá yðar sem syndlaus er kasti fyrsta steininum.	Let him who is without sin cast the first stone	let him who is without sin cast the first stone
Þú verður að fylgja henni heim.	Youve got to see her home.	youve got to see her home
Ég gæti haldið endalaust áfram um það en ég ætla það ekki	I could go on and on about it but I wont	i could go on and on about it but i wont
Hvað langar þig í eftirrétt	What would you like for dessert	what would you like for dessert
Ég er að leita að ákveðnum gömlum manni	Im looking for a certain old man	im looking for a certain old man
Ég komst ekki sökum úrhellisins	I could not come because of the heavy rain	i could not come because of the heavy rain
Hann verður góður kennari	He will be a good teacher	he will be a good teacher.

Table 5.46: BLEU Score for Icelandic languages (Test)

BLEU-1	0.757289
BLEU-2	0.723129
BLEU-3	0.715041
BLEU-4	0.651539

```
src=[Ég flýtti mér á brautarstöðina til þess eins að komast að því að lestin var þegar farin], target=[I hurried to the station only to find that the train had
src=[Ég var svo fúll], target=[I was so unhappy], predicted=[i was so unhappy]
src=[Vinsamlegast baðaðu barnið], target=[Please give the baby a bath], predicted=[please give the baby a bath]
src=[Ég er alltaf til], target=[Im always ready], predicted=[im always ready]
BLEU-1: 0.805251
BLEU-2: 0.775527
BLEU-3: 0.764592
BLEU-4: 0.703748
test
src=[Ég fór í lystigarðinn í gær], target=[I went to the park yesterday], predicted=[i went to the park yesterday]
src=[Þú munt bráðlega venjast því að tala opinberlega], target=[You will soon get used to speaking in public], predicted=[you will soon get used to speaking in
src=[Mér kom ekki dúr á auga í nótt], target=[I didnt sleep a wink last night], predicted=[i didnt sleep a wink last night]
src=[Ég komst ekki sökum úrhellisins], target=[I could not come because of the heavy rain], predicted=[i could not come because of the heavy rain]
src=[Við erum frænkur], target=[She and I are cousins], predicted=[she and i are cousins]
src=[Tom er prestur], target=[Tom is a pastor], predicted=[tom is a pastor]
src=[Ég er strákur], target=[I am a boy], predicted=[i am a boy]
src=[Það var skemmtilegt], target=[It was amusing], predicted=[it was amusing]
src=[Mér finnst gaman að spila fótbolta], target=[I like to play soccer], predicted=[i like to play soccer]
src=[Hún virðist þekkja okkur], target=[He seems to know us], predicted=[he seems to know us]
BLEU-1: 0.757289
BLEU-2: 0.723129
BLEU-3: 0.715041
BLEU-4: 0.651539
```

Fig 5.23: Train and Testing Result for Icelandic language

The full feature of training and testing result has shown in figure 5.23.

5.3.10 Macedonian Language

In table 5.47 has shown the Source, Target and Predicted Sentences for the Macedonian Language for the testing dataset. The final BLEU score for the testing session has given in table 5.48. The model has not given good translation quality for the Macedonian language. The final BLEU score is 0.308605. The overall result has shown in figure 5.24 for training and testing.

Table 5.47: Source, Target and Predicted Sentences for Macedonian Language (Test)

Source	Target	Predicted
Том збесна	Tom got furious	tom became furious
Можеме да се обидеме	We can try	we can try
Ќе биде ли Том спремен	Will Tom be ready	will tom be ready
Слави ли Том	Is Tom celebrating	is tom celebrating
Влегувај внатре	Get inside	get inside
Слушнавме пукот	We heard a gunshot	we heard a gunshot
Донеси му вода на Том	Get Tom some water	get tom some water

Table 5.48: BLEU Score for Macedonian language (Test)

BLEU-1	0.653591
BLEU-2	0.582543
BLEU-3	0.516964
BLEU-4	0.308605

A large amount of datasets are used for training the Macedonian language.But the accuracy of translation quality is the worst among all other languages.To get the best translation more and more training with new sentence pairs are needed.

```
src=[Том ја прати Мери да си оди], target=[Tom sent Mary away], predicted=[tom sent mary away]
src=[Том е влијателен], target=[Tom is influential], predicted=[tom is influential]
src=[Следи го], target=[Follow him], predicted=[follow him]
src=[Ќе се вратам во шест и пол], target=[Ill return at 630], predicted=[ill return at 630]
src=[Колку си беден], target=[Youre so pathetic], predicted=[youre so pathetic]
src=[Дома сум], target=[Im at home], predicted=[im home]
src=[Спушти го тој пиштол], target=[Put down that gun], predicted=[put down that gun]
BLEU-1: 0.674925
BLEU-2: 0.605221
BLEU-3: 0.537066
BLEU-4: 0.323117
test
src=[Том збесна], target=[Tom got furious], predicted=[tom became furious]
src=[Можеме да се обидеме], target=[We can try], predicted=[we can try]
src=[Ќе биде ли Том спремен], target=[Will Tom be ready], predicted=[will tom be ready]
src=[Треба да тргам], target=[I should get going], predicted=[i should get going]
src=[Слави ли Том], target=[Is Tom celebrating], predicted=[is tom celebrating]
src=[Том сите ги нервира], target=[Tom bugs everyone], predicted=[tom bugs everyone]
src=[Влегувај внатре], target=[Get inside], predicted=[get inside]
src=[Слушнавме пукот], target=[We heard a gunshot], predicted=[we heard a gunshot]
src=[Донеси му вода на Том], target=[Get Tom some water], predicted=[get tom some water]
src=[Тој ми е стар пријател], target=[Hes my old friend], predicted=[he my my old]
BLEU-1: 0.653591
BLEU-2: 0.582543
BLEU-3: 0.516964
BLEU-4: 0.308605
```

Fig 5.24: Train and Testing Result for Macedonian language

5.3.11 Kabyle Language

In table 5.49 has shown the Source, Target and Predicted Sentences for Kabyle Language in testing. The BLEU score for the language has shown in table 5.50.Table.

Table 5.49: Source, Target and Predicted Sentences for Kabyle Language (Test)

Source	Target	Predicted
Ssizdeg taxxamtim	Clean up your room	clean up your room
Tom yegguraɛd	Tom burped	Tom burped
Mxellen	Theyre crazy	theyre crazy
Ilaq ad iyitamneḍ	You must believe me	you must believe me
D iwessaren	Theyre old	theyre old
Sukkd afrannniḍen	Make another choice	make another choice
Agaṭunkent d aẓidan	Your cake is delicious	your cake is delicious

Table 5.50: BLEU Score for Kabyle language (Test)

BLEU-1	0.648385
BLEU-2	0.585603
BLEU-3	0.543434
BLEU-4	0.408951

In these research datatset are rich for kablye language.But according to the datasets the translation quality should be improved more. The overall result for training and testing has shown in the figure 5.25.

```
Source=[Ad mdyecnu Tom], Target=[Tom will sing for you], Predicted=[tom will sing for you]
Source=[Andat ttbut], Target=[Wheres the proof], Predicted=[wheres the proof]
Source=[D acu i nezmer ad kenttid nexdem], Target=[What can we do for you], Predicted=[what can we do for you]
Source=[Amek i sqqaṛen i babam], Target=[Whats your dads name], Predicted=[whats your dads name]
Source=[Ur ttɛaqabet ara Tom ɣef ayen], Target=[Dont punish Tom for that], Predicted=[dont punish tom for that]
Source=[Xedmeɣam kullec], Target=[I did everything for you], Predicted=[i did everything for you]
BLEU-1: 0.686006
BLEU-2: 0.623655
BLEU-3: 0.577418
BLEU-4: 0.438545
Test
Source=[Ssizdeg taxxamtim], Target=[Clean up your room], Predicted=[clean up your room]
Source=[Tom yegguraɛd], Target=[Tom burped], Predicted=[tom burped]
Source=[Mxellen], Target=[Theyre crazy], Predicted=[theyre crazy]
Source=[Tella kra n yiwet i yukren apaspuṛiw], Target=[Someone stole my passport], Predicted=[someone stole my passport]
Source=[Acḥal ara teseeddimt da], Target=[How long will you stay here], Predicted=[how long will you stay here]
Source=[D iwessaṛen], Target=[Theyre old], Predicted=[theyre old]
Source=[Sukkd afrannniḍen], Target=[Make another choice], Predicted=[make another choice]
Source=[Tuɣ yekkat wedfel], Target=[It was snowing], Predicted=[it was snowing]
Source=[Agaṭunkent d aẓidan], Target=[Your cake is delicious], Predicted=[your cake is delicious]
Source=[Ilaq ad iyitamneḍ], Target=[You must believe me], Predicted=[you must believe me]
BLEU-1: 0.648385
BLEU-2: 0.585603
BLEU-3: 0.543434
BLEU-4: 0.408951
```

Fig 5.25: Train and Testing Result for Kablye language

5.3.12 Tagalog Language

In table 5.51 has shown Source, Target and Predicted Sentences for Tagalog Language.The BLEU score for the testing dataset has given in table 5.52. The overall result for training and testing has shown in the figure 5.26 below.

Table 5.51: Source, Target and Predicted Sentences for Tagalog Language (Test)

Source	Target	Predicted
Hindi ka kayang tulungan ni Tom	Tom cant help you	tom cant help you
Sinong may alam	Who knows	who knows
Panoorin niyo ako	Watch me	watch me
Ano ang nasa pupitre	What is on the desk	what is on the desk
Tumakbo siya	He ran	he ran
Talagang di ako makalangoy	I cant swim at all	i cant swim at all
Lumapit ka	Come closer	come closer

Table 5.52: BLEU Score for Tagalog languages (Test)

BLEU-1	0.632754
BLEU-2	0.589577
BLEU-3	0.582804
BLEU-4	0.506442

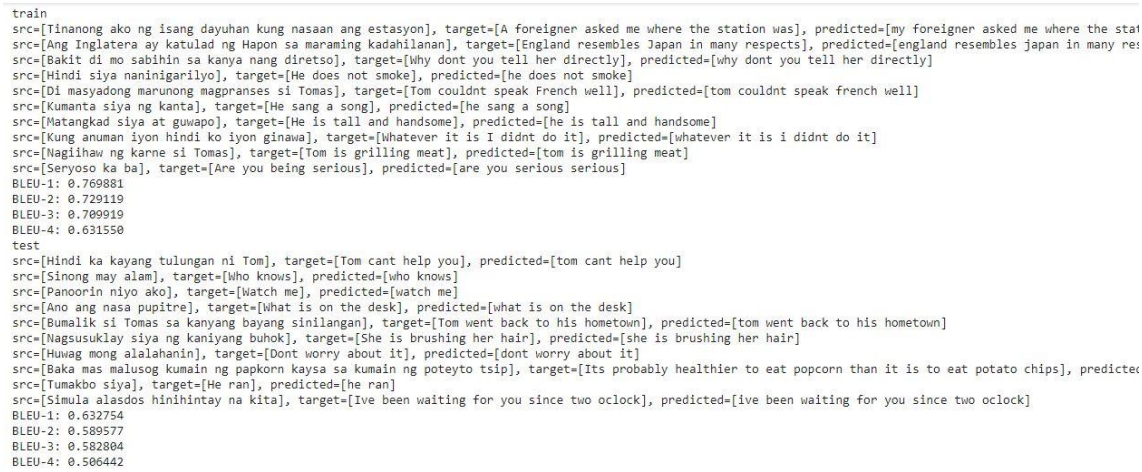


Fig 5.26: Train and Testing Result for Tagalog language

5.3.13 Cantonese Language

In table 5.53 we have shown the Source, Target and Predicted Sentences for Cantonese Language and table 5.54 have shown the BLEU score. The overall training result has shown in figure 5.27.

Table 5.53: Source, Target and Predicted Sentences for Cantonese Languages (Test)

Source	Target	Predicted
佢哋受過訓練。	Theyve been trained	theyve been trained
啲朱古力蛋糕隨便食喎。	Please help yourself to the chocolate cake	please help yourself to the chocolate cake
你介唔介意講大聲少少呀？	Would you mind speaking a little louder	would you mind speaking a little louder
實驗會唔會成功呢？	Will the experiment succeed	will the experiment succeed
我又結咗婚喇。	I got married again	i got married again
我個仔仲未識計加數。	My boy cant do addition properly yet	my boy cant do addition properly yet
佢上個禮拜病咗。	He was sick last week	he was sick last week

Table 5.54: BLEU Score for Cantonese languages (Test)

BLEU-1	0.747087
BLEU-2	0.718129
BLEU-3	0.713691
BLEU-4	0.653991

```
src=[我哋要請一個新嘅經理。], target=[We need to hire a new manager], predicted=[we need to hire a new manager]
src=[你而家係大人喺嘞喇，唔好再好似成個細路咁喇。], target=[Now that you are grown up you must not behave like a child], predicted=[now that you are grown up you must
src=[試吓從我個角度去諗吖。], target=[Try to look at it from my point of view], predicted=[try to look at it from my point of view]
src=[Tom係我哥哥。], target=[Tom is my older brother], predicted=[tom is my older brother]
src=[我等咗我個friend成粒鐘。], target=[Ive spent an entire hour waiting for my friend], predicted=[ive spent an entire hour waiting for my friend]
src=[你咁嘅成績入唔到大學喺喎。], target=[With those results you wont be able to go to university], predicted=[with those results you wont be able to go to universi
BLEU-1: 0.818451
BLEU-2: 0.787867
BLEU-3: 0.776278
BLEU-4: 0.718275
test
src=[佢哋受過訓練。], target=[Theyve been trained], predicted=[theyve been trained]
src=[啲朱古力蛋糕隨便食喎。], target=[Please help yourself to the chocolate cake], predicted=[please help yourself to the chocolate cake]
src=[你介唔介意講大聲少少呀？], target=[Would you mind speaking a little louder], predicted=[would you mind speaking a little louder]
src=[實驗會唔會成功呢？], target=[Will the experiment succeed], predicted=[will the experiment succeed]
src=[我哋食完晚飯，不如去海灘散步吖。], target=[Lets walk on the beach after dinner], predicted=[lets walk on the beach after dinner]
src=[我又結咗婚喇。], target=[I got married again], predicted=[i got married again]
src=[我個仔仲未識計加數。], target=[My boy cant do addition properly yet], predicted=[my boy cant do addition properly yet]
src=[佢返到嚟嗰陣，我啱啱好寫完封信。], target=[I had just written the letter when he came back], predicted=[i had just written the letter when he came back]
src=[聖誕節絕對係我最鍾意嘅節日。], target=[Christmas is definitely my favorite holiday], predicted=[christmas is definitely my favorite holiday]
src=[佢上個禮拜病咗。], target=[He was sick last week], predicted=[he was sick last week]
BLEU-1: 0.747087
BLEU-2: 0.718129
BLEU-3: 0.713691
BLEU-4: 0.653991
```

Fig 5.27: Train and Testing Result for Cantonese language

5.3.14 Low German Language

Low German languages is another language which translation of English could not find in google translator.The established neural translation model worked with low german languages datasets.Due to scarcity of the datatsets the training and testing session with these language is difficult.Further research should be done for accurate translation.The research has paved the way translation with uncommon languages.In table 5.55 have shown the Source, Target and Predicted Sentences for Cantonese Language and table 5.56 have shown the BLEU score. The overall training result has shown in figure 5.28.

Table 5.55: Source, Target and Predicted Sentences for Low German Language (Test)

Source	Target	Predicted
Is se weg	Is she gone	is she gone
Mien leevsten Sport is Football	My favorite sport is soccer	my favorite sport is soccer
Ik heff keen Hoor op ’n Kopp	I have no hair on my head	i have no hair on my head
Wo kennst du mien Naam von	How do you know my name	how do you know my name
Mien Rügg deit jümmer noch weh	My back still hurts	my back still hurts
Güstern heff ik en Book köfft	Yesterday I bought a book	yesterday i bought a book
Se weren all so mööd dat se anners nix as jappen kunnen	They were all so tired that they could do nothing but yawn	they were all so tired that they could do nothing but yawn

Table 5.56: BLEU Score for Low German language (Test)

BLEU-1	0.749318
BLEU-2	0.713802
BLEU-3	0.702898
BLEU-4	0.630855

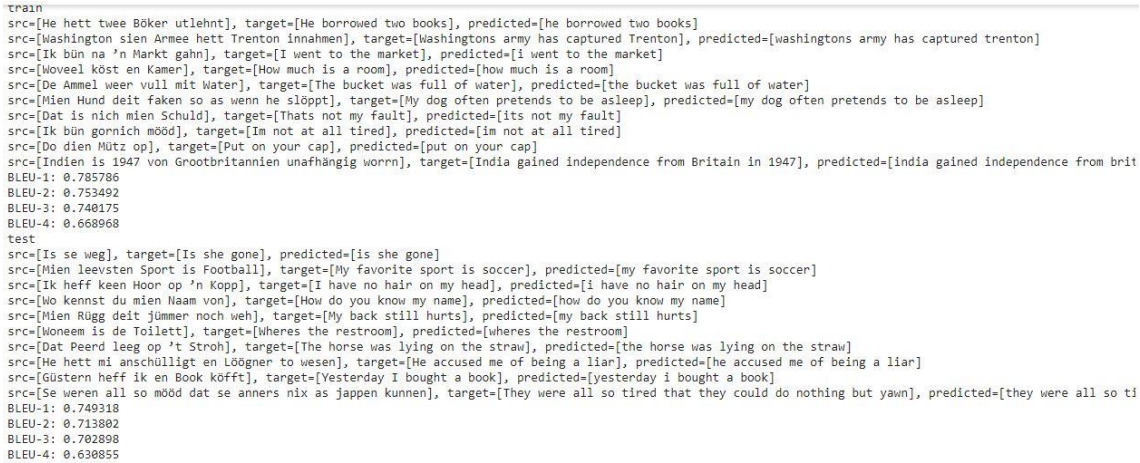


Fig 5.28: Train and Testing Result for low German language

5.4 COMPARISON BY BLEU SCORE

To evaluate the translation in our research the most popular matrix BLEU have used. BLEU, NIST, METEOR, and TER metrics are used most often in the evaluation of MT quality. Despite the well-known problems with BLEU, and the availability of many other metrics, MT system developers have continued to use BLEU as the primary measure of translation quality. In this section, the BLEU score of Google Translator the most popular machine translation system and our model BLEU score have discussed Translation accuracy of not available languages also have shown the.At last section the accuracy of some common languages have shown.

5.4.1 Comparison for Languages Available in Google Translation

First, the translation quality of the languages available in google translation have improved by using established Neural Machine Translation.In fact among them some of the languages have given better translation quality than the Google translator. The BLEU score of different language pairs have found in two different research papers which have worked with the Google translation accuracy. In the thesis paper [48] the BLEU score of Google translator for translating different languages to English have manipulated.Then Google Translator used Statistical Machine Translation for

translation. According to the Aiken, M., & Balan, S.[48] the comparison of the BLEU score of different language for translating into English with Google translator and our languages is given below.

Table 5.57: Comparison the BLEU Score Between our proposed model and GT

Language	Our Model BLEU Score	Google Translate BLEU score
Bengali	0.825149	N/A
Romanian	0.862303	0.55
Danish	0.821216	0.93
Czech	0.866760	0.58
Arabic	0.565170	0.68
Thai	0.368639	0
Marathi	0.404108	N/A
Vietnamese	0.528193	0.13
Icelandic	0.647203	0.48
Macedonian	0.308605	0.15

Here except for the Bengali, Thai and Marathi languages BLEU score for Google translator of other languages have shown. Comparing with the SMT, our model has produced a pretty good BLEU score. Mostly, for Macedonian language or Vietnamese language, the established model has given excellent result comparing with Google translatorHere another matter has revealed.According to the table 5.57 in the era of SMT google languages translation accuracy was poor.In another thesis paper, the updated BLEU score of Google translator have found. In 2016 then Google Translator has entered in the neural machine translation system. The paper [49] has depicted the updated Google Neural Machine Translation BLEU score which has indicated excellent development in the BLEU score for different languages.These indicates the Excellency of using neural machine translation model for translation. Bengali language has own grammatical structure and large vocabulary.The letters and scripts that is used by the begali language is also unique.So,the translation of Bengali to English is difficult one for machine to fully understand the context and meaning of the sentence.

Table 5.58: Comparison the BLEU Score Between our proposed model and GNMT

Language	Our Model BLEU Score	GNMT BLEU score
Bengali	0.825149	N/A
Romanian	0.862303	0.84
Danish	0.821216	0.82
Czech	0.866760	0.86
Arabic	0.565170	0.76
Thai	0.368639	0.68
Marathi	0.404108	N/A
Vietnamese	0.528193	0.72
Icelandic	0.647203	0.72
Macedonian	0.308605	0.61

Along with the new Google translator accuracy established model has not given so much good BLEU score as the Google Translator has given now. But according to the BLEU score interpretation of figure 5.29 that has given proposed model translation quality is good for most of the languages.

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

Figure 5.29: BLEU score interpretation

If a table according to the interpretation of the BLEU score has made the translation quality of our model have been fully understood. Figure 5.59 have shown the interpretation of translation quality of established model.

Table 5.59: Translation quality of the 10 languages according to BLEU score

Language	Our Model BLEU Score	Interpretation of the translation quality
Bengali	0.825149	Quality often better than human
Romanian	0.862303	Quality often better than human
Danish	0.821216	Quality often better than human
Czech	0.866760	Quality often better than human
Arabic	0.565170	Very high quality, adequate and fluent translations
Thai	0.368639	Understandable to good translations
Marathi	0.404108	High quality translations
Vietnamese	0.528193	Very high quality, adequate and fluent translations
Icelandic	0.647203	Quality often better than human
Macedonian	0.308605	Understandable to good translations

The Neural machine translation has given pretty good translation among the languages for translating into English. Romanian languages and English languages have the similarity of their scripts. Both are used in Latin scripts and the same alphabets. For both in our model and Google translator, the translation quality for Romanian, Danish, Czech languages is higher than any other languages. On the other hand Thai, Arabic, Marathi, Vietnamese, Macedonian and Icelandic have given good translation quality but not the best. Established machine translation model has given the translation only from different languages to English.

5.4.2 Comparison for Few Common languages

13 different languages have also used to translate into English. These are Turkish, German, French, Swedish, Portuguese, Spanish, Japanese, Greek, Bulgarian, Finish, Indonesian, Azerbaijani and Norwegian. The established model also worked well with these languages. For these languages, Google translator has already given the best translation quality. For these reasons, our research has not analyzed deeply by using these language datasets. In table 5.60 have shown the BLEU score for these languages in the established model. Only 100 epochs is used for every translation. Further training has not done for these languages.

Table 5.60: Translation quality of the common languages according to BLEU score

Language	Our model BLEU score	Interpretation of the translation quality
Turkish	0.434650	High Quality Translations
German	0.478521	High Quality Translations
Swedish	0.33986	Understandable to good translations
French	0.478799	High Quality Translations
Portuguese	0.471336	High Quality Translations
Spanish	0.537467	Very high quality, adequate and fluent translations
Japanese	0.387811	Understandable to good translations
Greek	0.353354	Understandable to good translations
Bulgarian	0.493094	High Quality Translations
Finish	0.600799	Very high quality, adequate and fluent translations
Indonesian	0.765196	Quality better than human
Azerbaijani	0.339542	Understandable to good translations
Norwegian	0.383382	Understandable to good translations

Research should be done with those languages in which translation quality needs to improve a lot. Here among these 13 languages, most of the languages have given excellent translation quality in Google translator. Spanish, French, Turkish, Portuguese, Indonesian translation BLEU score is also very high. So, for this reason, the research does not focus on these languages. But our model has also worked better for these languages. According to the most popular BLEU matrix interpretation, our model is doing the translation accurately. A little more research has shown these to indicate that the model tuning and capability to translate different languages to English is satisfactory. The model has improved a lot. More training and using higher datasets may be given the best translation quality of these languages also.

5.4.3 Accuracy for Languages Not Available in Google Translation

The established neural translation model has also worked with 4 uncommon languages for translating English. Google translator does not even add these languages to their system. This language translation is not available on the Google translator Platform. These four languages are Kablye, Tagalog, Low German and Cantonese. Among popular translators, Yandex Translator has added Tagalog for translation.

Microsoft Bing translator has added the Cantonese language for translation. But the other two language kablye and Low German has not been added any popular translator. The translation to the English language from these languages is difficult because of this. The established model has used to translate these four languages into English. The BLEU score for these languages and the interpretation of the BLEU score have given below.

Table 5.61: Translation quality of the uncommon languages according to BLEU score

Languages	BLEU Score	Interpretation of the translation quality
Kabyle	0.408951	High quality translations
Tagalog	0.506442	High quality translations
Low Saxon	0.630855	Quality often better than human
Cantonese	0.653991	Quality often better than human

In table 5.61 have shown the translation quality for these languages according to BLEU score. By doing proper training this result has gotten and indicates the model ability to translate language properly.

5.5 COMPARISON WITH EXAMPLE

Today the most used and popular translator is Google translator. So, a comparison has been made between Google translator and our established neural machine translation model. Sentence by sentence comparison has been made. So, by doing this comparison some surprising results have been found. Our proposed model has given better translation than existed translation system for different sentences. The BLEU score and the translation quality for 10 languages have shown and discussed. These sections have shown the disparities of output of Google translate and established neural machine translation. First, the comparison between Our Neural Machine Translation Model and Google Translator for the Bengali Languages has seen in table 5.62. Dissimilarities among different sentences have shown also. Established model translation quality is better for many translations than the Google Translator

Table 5.62: Comparison between NMT and Google Translator for Bengali languages

Bengali	NMT	Google Translate
এটা নিজের ঘর মনে করো।	make yourself at home	Think of it as your own home.
টম সত্যি বাক্যবাগীশ।	tom certainly is eloquent.	Tom is really eloquent.
টম বারবার দরজায় ধাক্কা দিলেন কিন্তু কেউ বেরোলেন না।	tom kept knocking on the door but nobody came.	Tom repeatedly knocked on the door but no one came out.
টম খুব মন দিয়ে শুনছে বলে মনে হচ্ছে|	tom seems to be listening carefully	Tom seems to be listening intently
আমি সবে বাড়ি পৌছালাম।	i just got home	I barely got home.
আমি আমার পাটা জিরোচ্ছে। আমি আমার পা টা জিরোচ্ছে।	im resting my legs	I'm paying my dues. I'm kicking my leg.
আপনি আমার প্রতি বড্ড অধৈর্য্য।	youre so impatient with me	You are impatient with me.
টম চূড়ান্তভাবে অপদস্ত হোল।	tom was utterly humiliated	Tom is finally humiliated.
তুই আমার রাস্তা আটকে রেখেছিস।	you are in my way	You blocked my way.
আমার মা চটপটে।	my mother is active.	My mother is agile.
আমাদের ডাকিস	call us	Our ducks.
আপনি পিয়ানোটা তুলতে পারবেন না।	you cant lift the piano	You can't pick up the piano.

For Example “আমি সবে বাড়ি পৌছালাম”, for this, the established neural machine translation has given “I just got home”. In Google translator it has given “I barely got home.” So, to compare these two translations the dissimilarities has pointed. Another is “আমাদের ডাকিস” here we for these sentence the result for established Neural translation model is “call us”. But in Google translator gives “Our ducks” which is fully opposite of the meaning of the sentence. Not all sentences but some sentences give better result in the established neural translation model. Another example is when rain tree is translated to Bengali by using any translator the output is "বৃষ্টি গাছ". But rain tree is called "শিরীষ গাছ" in Bengali. The translator system cannot give this translation. Machine translation system simply translates the language based on straight meaning. This is one of the

biggest lagging that found in the machine translation system. Table 5.63 has shown the comparison for the Romanian language

Table 5.63: Comparison between NMT and Google Translator for Romanian languages

Romanian	NMT	Google Translate
nu intarzia la scoala	dont be late for school	he wasn't late for school
il evit pe tom	im avoiding tom	I avoid tom
mia fost frig	i felt cold	I was cold
stia ca nu va putea castiga	he knew he could not win.	he knew he would not be able to win
doar falsifical	just fake it	only falsifical
cineva trebuie sa fie tras la raspundere	someones got to be held accountable	someone must be held accountable
tom nu a mai zburat cu avionul	tom has never been on a plane before	tom didn't fly anymore

The established neural translation model gives a better BLEU score than the Google translator for the translation of the Romanian to English. Just as “il evit pe tom” the translation got in neural machine translation is “im avoiding tom”. And Google translator gives “I avoid tom”. Although, there is not a big difference in translation there are slight differences and problems with the tense.

Table 5.64: Comparison between NMT and Google Translator for Danish languages

Danish	NMT	Google Translate
tom vil maske vidne	tom might testify	tom will mask witness
jeg er en smule sulten	im a bit hungry	I'm a little hungry
hun var nedtrykt	she felt blue	she was depressed
tom er endnu ikke teenager	tom isnt a teenager yet.	tom is not yet a teenager
vgtstngerne er tunge	the barbells are heavy.	the weights are heavy
jeg vil gerne tjekke ud nu	i want to check out now	I would like to check out now
det er hvad jeg sa	thats what i saw	that's what i said

In figure 5.64 the comparison for the Danish languages.Consider examples for the translation of Danish to English in this machine translation system “vgtstngerne er tunge” the translation of this sentence in NMT is “the barbells are heavy.”.But in the Google translator gives “the weights are heavy”. There also have shown a difference in

grammar. “jeg vil gerne tjekke ud nu” the translation is “I want to check out now” in established NMT and Google translator, the translation is “I would like to check out now”. The established translation model has given a better result.

Table 5.65: Comparison between NMT and Google Translator for Czech languages

Czech	NMT	Google Translate
deti prichazi	here come the children	children are coming
tohle dopis	finish this	this letter
tom mi jeden poslal	tom sent one to me	one sent me that
Projizdim	im getting by	I drive
chci odejit	i want to quit	I want to leave
libi se mi tu	i like being here	I like it here
nebude to pro nas snadne	it wont be easy for us	it will not be easy for us

Table 5.65 has shown another comparison between NMT and Google translator for the Czech language. The translation “tohle dopis” for this have got the English translation for NMT is “finish this”. But in GT produces “this letter” which contradicts the meaning of the NMT. “tom mi jeden poslal” English translation is “ tom sent one to me” but in the google translator gives “one sent me that”. In table 5.66 have shown the comparison for the Arabic language.

Table 5.66: Comparison between NMT and Google Translator for the Arabic language

Source	NMT	Google Translate
تحرك عقرب الثواني الخاص بالساعة .	the clock ticked	The watch's second hand moved
التمس الحذر وأنت تعبر الطريق.	be careful when you cross a road	Seek caution as you cross the road.
أحب معاملتك لي.	i like the way you treat me	I like you treat me.
لقد ارتكبت الكثير من الأخطاء.	you have made many mistakes	I made a lot of mistakes.
لست سريعاً بما يكفي.	you're not fast enough	I am not fast enough.
أدرس ثلاث ساعات في اليوم.	i study for 3 hours every day	I study three hours a day.
سأعود لآخذ حقيبة يدي.	i'll return to get my handbag.	I’ll come back to pick up my handbag.

As for the Arabic languages there showed some results that are better than the Google translator. For example, consider the first sentence “ تحرك عقرب الثواني الخاص بالساعة.”Here

in Google translator, the result is “The watch's second hand moved”. But on the other hand for neural machine translation, translates, “the clock ticked” which is more simple and clear. The difference of the tense and person has identified in the translation. All in all some translation gives better result in this established neural machine translation.Table 5.67 is for the Thai language sentence comparison between our model and Google translator.

Table 5.67: Comparison between NMT and Google Translator for the Thai language

Source	NMT	Google Translate
คุณสามารถอยู่ได้จนถึงคืนนี้	you can stay till tonight	You can stay until tonight
พวกเขาทำลายชีวิตของฉัน	they ruined my life	They ruin my life.
ทอมอาจจะรู้จักแมรี่	tom might know mary	Tom may know Mary.
พี่ชายของฉันไปเป็นพ่อครัว	my brother became a cook	My brother went to cook.
จินตนาการของทอมถูกกระตุ้น	tom's imagination was arouse	Tom's imagination was stimulated
ฉันอยากเรียนที่ปารีส	i'd like to study the paris	I want to study in Paris.
ตอนนั้นฉันอยู่ในเขา	i was in the mountains.	At that time i was in him.

Here, Some problems with present and past tense translation have got which differentiate translation between established neural machine translation and Google translator which is different from the other translation.

For Marathi the comparison table 5.68 is given below.

Table 5.68: Comparison between NMT and Google Translator for the Marathi language

Marathi	NMT	Google Translate
तुला कळणार नाही	You will not know	you wont understand.
त्याला हात लावू नका	dont touch it	Don't touch him
माझ्याकडे नव्हतं	i didnt have it	I didn't have
मला हवं आहे	i want it	I want
मी चालत जाईन]	i will go on foot	I will walk
माझा बोका गायब झाला आहे	my cat is missing	My goat has disappeared
आकर्षक नाहीये	it isnt attractive	Not attractive.

Here, the odd result of Google translator has identified. For example here "माझा बोका गायब झाला आहे" for this sentence the established neural machine translation model gives "my cat is missing". But for Google translate it gives the translation "My goat has disappeared". "त्याला हात लावू नका" for this sentence the translation for established neural translation model gives "don't touch it". But as for Google translat gives the result "Don't touch him". So, the meaning of the translation also has changed. For the Vietnamese language, the comparison is shown in Table 5.69.Our proposed model result and Google translates result for Vietnamese languages are better than the other translator.

Table 5.69: Comparison between NMT and Google Translator for Vietnamese languages

Vietnam	NMT	Google Translate
Ấy mua bộ đồ này ở đâu thế	where do you buy clothes	Where did you buy this suit
Đã đến lúc trả đũa rồi	its payback time	Time to retaliate
Tom để ý thấy cánh cửa chỉ khép hờ	tom noticed the door was half closed	Tom noticed that the door was only partially closed.
Trong hai tuần nữa bạn sẽ có thể ra viện	in another two weeks you will be able to get out of the hospital	In two weeks you should be able to leave the hospital
Cô ấy thích nó	she liked that	She likes it
Tôi đã quyết định là kể từ lúc này tôi sẽ học chăm chỉ hơn	ive made up my mind to study harder from now on	I decided that from now on I would study harder.

Vietnamese language almost has given the same result but in some aspects, the output is not the same. For example there one of the sentence "Tôi đã quyết định là kể từ lúc này tôi sẽ học chăm chỉ hơn".For this sentence, the Google translate result is "I decided that from now on I would study harder". But our proposed model result is "ive made up my mind to study harder from now on". Also for some sentences, Google translator has given the complex translation of the meaning of the context of the sentences.

Table 5.70: Comparison between NMT and Google Translator for Icelandic languages

Icelandic	NMT	Google Translate
Þetta var málið	That hit the spot	That was the point
Meer finest egg vend	i dislike eggs	I find eggs bad
Dreptu á dyrnar	knock on the door	Kill the door
Það er ekki hag að sutra incur í closeting	the toilet doesnt flush	It is not possible to flush down the toilet
Dragið ekki dár að útlendingum	dont poke fun at foreigners	Do not draw strangers to foreigners
Hún hefur fengið góða menntun	she has received a good education	She has a good education
Sá yðar sem syndlaus er kasti fyrsta steininum.	let him who is without sin cast the first stone	He that is without sin among you, let him first cast a stone.
Hann var mér reiður vegna þess að ég sagði honum upp	He was mad at me because I broke up with him.	He was angry with me because I fired him

For Icelendic language table 5.70 has given the comparison of our model and Google translator.The Icelandic sentence is “Hún hefur fengið góða menntun”. For these, NMT has given English translation “she has received a good education” but in Google translator, have found “She has a good education”.

Table 5.71: Comparison between NMT and Google Translator for Macedonian languages

Macedonian	NMT	Google Translate
Том простенка	tom groaned	Tom’s sorry
Прва година сум	im just a freshman	This is my first year
Угаси го огнот	put that fire out	Extinguish the fire
Ќе наминам попосле	ill stop by later	I will leave later
Том изгледа сомнително	tom looks dubious	Tom looks suspicious
Одам јас во теретана	im off to the gym	I’m going to the gym
Го оставам Том да спие до подоцна	i let tom sleep in	I leave Tom to sleep until later
Татко ми има работа	my father is busy	My father has a job

In table 5.71 has shown the comparison of sentences for the Google translator and our model. Here some indifference have also found between our model translation and

Google translator. For example, the sentence is “Том простенка” .The established model gives “tom groaned”. But in Google translator translates “Tom’s sorry”. For another sentence “Угаси го огнот” to translate in English “put that fire out” has given by established NMT. But the Google translator has given the translation “Extinguish the fire”.

5.6 SUMMARY

In this research, 27 different languages datasets are used. For 14 languages the analysis has gone through far. In these 14 languages analyzing, comparing the research has revealed some conclusion. More epochs are used for better translation. Another thing is that scripts that are similar to the English language often get a better result than any other scripts. Here with the 14 languages have analyzed further for translation. And for the uncommon languages, has given high translation quality. This indicates the established model skills for translation. BLEU score gives reliable translation quality measurement. Although sentence to sentence comparison has also done so that anyone can understand the problem that is associated with machine translation system nowadays.

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

This chapter describes the conclusion and future research destination of this thesis work. First the overall brief of this thesis work is presented in the conclusion section. Then the potential future research way is provided in planned future works section.

6.2 CONCLUSION

Neural machine translation models fit a single model rather than a pipeline of fine-tuned models and currently achieve state-of-the-art results. Machine translation is the task of automatically converting source text in one language to text in another language. Although effective, the neural machine translation systems still suffer some issues, such as scaling to larger vocabularies of words and the slow speed of training the models. There are the current areas of focus for large production neural translation systems, such as the Google system. In this research paper neural machine translation is occurred to translate many languages to English. Mostly here 14 languages are used for translating in English. Among them 4 languages are uncommon to translate in English. This research leads the path to work with the regional languages and many other languages that are not very well used in the machine translation system. The research paves the way for working more and more with neural machine translation system in language translation so that the improvement of machine translation can go further comparing with the human evaluation. There is no doubt that Google translator, Microsoft Bing translator and other different types of translator have done a good translation and their translation quality has improved day by day. But every translator has their laggings.

This is that most of the time translation system cannot get the contents of the sentence or words. They are simply translating based on some rules or regulation. All in all, machine translation cannot understand the tune of any languages whenever translates from one language to another. Some examples of not getting the vibes of any languages have also shown in this research. As Latin scripts have used in English so which languages have used the Latin scripts have better translation quality than other languages that have used different scripts. The Latin scripts language translation accuracy is better than the non-Latin scripts language. The researchers have proved that Asian languages translation quality needs to improve more. Most Asian language uses non-Latin scripts. The whole research has given some conclusion. These are given below.

i. The research has worked with 4 languages whose English translation is not available in google translator. The average BLEU score is found 0.50. The score indicates the high translation quality.
ii. Another translation of 10 different languages to English is also shown. The average BLEU score of these languages are 0.51769674 indicates a very good quality translation.
iii. 14 common languages have also used which translation quality in the existing system is good. The established machine translation system has also shown better result for these languages.
iv. The Latin scripts language translation accuracy is better than the non-Latin scripts language. The researchers have proved that Asian languages translation quality needs to improve more. Most Asian language uses non-Latin scripts.

6.3 PLANNED FUTURE WORKS

In future, with larger datasets, the accuracy will be better and by using this neural translation model. Different datasets should be used for every language for evaluating the translation quality better. Also, human evaluation can be concluded for validating the translation quality more. Here, Punctuation problems should be solved in a planned way. Bi-directional LSTM can be used for further improvement.

LIST OF REFERENCES

[1] Ilya Pestov.2018.A history of machine translation from the Cold War to deep learning.(online). #MACHINE LERANING.https://www.freecodecamp.org/news/a-history-of-machine-translation-from-the-cold-war-to-deep-learningf1d335ce8b5/(March 12,2018).

[2] Galley, M. and Manning, C.D. October 2008. A simple and effective hierarchical phrase reordering model. *In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* pp. 848-856.

[3] Koehn, P., Axelrod, A., Mayne, A.B., Callison-Burch, C., Osborne, M. and Talbot, D. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. *In International Workshop on Spoken Language Translation (IWSLT)*.

[4] Chiang, D. (2007). Hierarchical phrase-based translation. *computational linguistics*. **33**(2): 201-228.

[5]Bahdanau, D., Cho, K. and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate . *arXiv preprint arXiv:1409.0473*.

[6]Popovic, M. and Ney, H. May, 2006. POS-based Word Reorderings for Statistical Machine Translation . *In LREC.* pp. 1278-1283.

[7] Crego, J.M. and Marino, J.B. December, 2006. Reordering experiments for n-gram-based smt. *In 2006 IEEE Spoken Language Technology Workshop.* pp. 242-245.

[8] Durrani, N., Schmid, H., Fraser, A., Koehn, P. and Schütze, H., 2015. The operation sequence model—combining n-gram-based and phrase-based statistical machine translation. Computational Linguistics, **41**(2): 185-214.

[9] Sutskever, I., Vinyals, O. and Le, Q.V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*. **27**:3104-3112.

[10] Schwenk, H. July 2004. Efficient training of large neural networks for language modeling. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*. **4**:3059-3064.

[11] Och, F.J. July 2003. Minimum error rate training in statistical machine translation. *In Proceedings of the 41st annual meeting of the Association for Computational Linguistics.* pp. 160-167.

[12] Hopkins, M. and May, J. July,2011.Tuning as ranking. *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing.* pp. 1352-1362.

[13] Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A.J., Koehn, P., Logacheva, V., Monz, C. and Negri, M. August 2016. Findings of the 2016 conference on machine translation. *In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. **2**:131-198.

[14] Stanojević, M. and Sima'an, K. October 2014. Fitting sentence level translation evaluation with many dense features. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* pp. 202-206.

[15] Lo, C.K. and Wu, D. June, 2011. MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. pp. 220-229.

[16] Bojar, O., Graham, Y., Kamran, A. and Stanojević, M. August 2016. Results of the wmt16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. **2**:199-231.

[17] Sennrich, R., Haddow, B. and Birch, A. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

[18] Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. and Makhoul, J. June, 2014. Fast and robust neural network joint models for statistical machine translation. *In proceedings of the 52nd annual meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. **1**:pp.1370-1380.

[19] Niehues, J., Cho, E., Ha, T.L. and Waibel, A. 2014. Analyzing neural MT search and model performance. *arXiv preprint arXiv:1708.00563*.

[20] Lee, J., Cho, K. and Hofmann, T. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*. **5**: 365-378.

[21] Sennrich, R., Haddow, B. and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909.*

[22] Och, F.J. and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*. **29**(1):19-51.

[23] Forcada, M.L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Sánchez-Martínez, F., Ramírez-Sánchez, G. and Tyers, F.M. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*. ***25***(2):127-144.

[24] Singh, A. and Singh, P. 2015. Punjabi dialects conversion system for Malwai and Doabi dialects. *Indian Journal of Science and Technology*. **8**(27):1-6.

[25]Bennett, W.S. and Slocum, J. 1985. The LRC machine translation system. *Computational linguistics*. **11**(2-3):111-121.

[26] Vogel, S., Zhang, Y., Huang, F., Tribble, A., Venugopal, A., Zhao, B. and Waibel, A. September, 2003. The CMU statistical machine translation system. *In Proceedings of MT Summit* . **9**:54-61.

[27]Milon, H.R., Sabbir, S.N.U., Inan, A. and Hossain, N. June 2020. A Comprehensive Dialect Conversion Approach from Chittagonian to Standard Bangla. *In 2020 IEEE Region 10 Symposium (TENSYMP)*. pp. 214-217. IEEE.

[28]Nusai, Chutchada, Yoshimi Suzuki, and Haruaki Yamazaki. 2008. Estimating word translation probabilities for Thai–English machine translation using EM Algorithm. *International Journal of Computational Intelligence*. **4** :3.

[29] Arefin, M.S., Hoque, M.M., Rahman, M.O. and Arefin, M.S. May, 2015. A machine translation framework for translating Bangla assertive, interrogative and imperative sentences into English. *In 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*. pp. 1-6.

[30] Alam, M.G.R., Islam, M.M. and Islam, N. 2011. A New Approach To Develop An English To Bangla Machine Translation System . *Daffodil International University Journal of Science and Technology*. **6**(1): 36-42.

[31] Mukta, A.P., Mamun, A.A., Basak, C., Nahar, S. and Arif, M.F.H. 2011. A Phrase-Based Machine Translation from English to Bangla Using Rule-Based Approach. *In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. pp.1-5.

[32] Francisca, J., Mia, M.M. and Rahman, S.M. 2011. Adapting rule based machine translation from english to bangle. *Indian Journal of Computer Science and Engineering (IJCSE)*. **2**(3): 334-342.

[33] Chowdhury, M.S.A. 2013. Developing a Bangla to English Machine Translation System Using Parts Of Speech Tagging: A Review. *Journal of Modern Science and Technology*. **1**(1):113-119,

[34] Ali, M. and Ali, M.M. 2002. Development of machine translation Dictionaries for Bangla language. *In 5th ICCIT*. pp.272-276.

[35] Siddique, S., Ahmed, T., Talukder, M.R.A. and Uddin, M.M. 2020. English to Bangla Machine Translation Using Recurrent Neural Network. *International Journal of Future Computer and Communication*. **9***(*2).

[36] Islam, M.Z., Tiedemann, J. and Eisele, A. May, 2010. English to Bangla phrase-based machine translation. *In Proceedings of the 14th Annual conference of the European Association for Machine Translation*.

[37] Salam, K.M.A., Yamada, S. and Nishino, T. 2013. How to translate unknown words for English to Bangla Machine Translation using transliteration . *Journal of computers*. **8**(5):1167-1174.

[38] Ali, M.N.Y., Al-Mamun, S.A., Das, J.K. and Nurannabi,A.M. 2008. Morphological analysis of bangla words for universal networking language. *In 2008 Third International Conference on Digital Information Management*. pp.532-537.

[39] Anwar, M.M., Anwar, M.Z. and Bhuiyan, M.A.A. 2009. Syntax analysis and machine translation of Bangla sentences. *International Journal of Computer Science and Network Security*. **9**(8):317-326.

[40] Muntarina, K., Moazzam, M.G. and Bhuiyan, M.A.A. 2013. Tense based english to Bangla translation using MT system. *International Journal of Engineering Science Invention*. **2**(10):30-38.

[41] Rabbani, M., Alam, K.M.R. and Islam, M. May, 2014. A new verb based approach for English to Bangla machine translation. *In 2014 International Conference on Informatics, Electronics & Vision (ICIEV)*. pp. 1-6.

[42]Roy, M. and Popowich, F. May, 2010. Word reordering approaches for bangla-english statistical machine translation. *In Canadian Conference on Artificial Intelligence Springer, Berlin, Heidelberg*. pp. 282-285.

[43] Nitta, Y. 1986. Problems of machine translation systems: Effect of cultural differences on sentence structure . *Future Generation Computer Systems*. **2**(2):101-115.

[44] Lyons, S. August, 2016. Quality of Thai to English Machine Translation. *In Pacific Rim Knowledge Acquisition Workshop (pp. 261-270).Springer,Cham*.pp.261-270.

[45] Saini, S. and Sahula, V. February 2015. A survey of machine translation techniques and systems for indian languages. In *2015 IEEE International Conference on Computational Intelligence & Communication Technology* . pp. 676-681.

[46] Antony,P.J. March 2013. Machine translation approaches and survey for Indian languages. *In International Journal of Computational Linguistics & Chinese Language Processing.***18**(1): pp. 47-78.

[47] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. and Klingner, J. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.0814*.

[48] Aiken, M., & Balan, S. 2011. An analysis of Google Translate accuracy. *Translation Journal*. **16**(2). Retrieved July 8, 2019.

[49] Milam Aiken. 2019. An Updated Evaluation of Google Translate Accuracy. *Studies in Linguistics and Literature*. **3**(3).

[50] Callison-Burch, C., Osborne, M. and Koehn, P. 2006, April. Re-evaluation the role of bleu in machine translation research. *In 11th Conference of the European Chapter of the Association for Computational Linguistics.*

[51] Firat, Orhan, Kyunghyun Cho, Baskaran Sankaran, Fatos T. Yarman Vural, and Yoshua Bengio. "Multi-way, multilingual neural machine translation." *Computer Speech & Language 45 (2017): 236-252.***45**: 236-252.

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

[53] Dong, Daxiang, Hua Wu, Wei He, Dianhai Yu,Haifeng Wang. 2015. Multi-task learning for multiple language translation. *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. **1**:1723-1732.

[54] Papineni, K., Roukos, S., Ward, T. and Zhu, W.J. July,2002. BLEU: a method for automatic evaluation of machine translation. *In Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. pp. 311-318.

[55] Md. Arid Hasan; Firoj Alam; Shammur Absar Chowdhury and Naira Khan. Neural Machine Translation for the Bangla-English Language Pair. *2019 22nd International Conference on Computer and Information Technology (ICCIT)*. IEEE.

[56]Baniata, Laith H., Seyoung Park, and Seong-Bae Park. 2018. A neural machine translation model for Arabic dialects that utilizes multitask learning (MTL). *Computational intelligence and neuroscience 2018 .*

[57]Luekhong, Prasert, Peerat Limkonchotiwat, and Taneth Ruangrajitpakorn. 2019. A study on an effect of using deep learning in Thai-English machine translation processes. *2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*. IEEE.

[58] Wei He, Zhongjun He,Hua Wu and Haifeng Wang. 2016. Improved neural machine translation with SMT features. *Proceedings of the AAAI Conference on Artificial Intelligence*. **30**(1).

[59] Jason Brownlee.(updated).2017.A Gentle Introduction to Calculating the BLEU Score for Text in Python.(online).Deep Learning for Natural Language Processing. https://machinelearningmastery.com/calculate-bleu-score-for-text-python (December 19,2019).

[60] Maučec, Mirjam Sepesy, and Gregor Donaj. 2019. Machine Translation and the Evaluation of Its Quality."*Recent Trends in Computational Intelligence*. Submitted: May 16th 2019 Reviewed: August 7th 2019 Published: September 7th 2019.

APPENDIX A

There has given the Bengali language code.All language have used the same code except the datasets. In every time when worked with a new language dataset form this language to English is used.That is the difference.

Code: Develop a Neural Machine Translation System

A-1: Preparing the Text Data

For Latin scripts languages

```
import string
import re
from pickle import dump
from unicodedata import normalize
from numpy import array

# load doc into memory
def load_doc(filename):
 # open the file as read only
  file = open(filename, mode='rt', encoding='utf-8')
  # read all text
  text = file.read()
  # close the file
  file.close()
  return text

# split a loaded document into sentences
```

```
def to_pairs(doc):
  lines = doc.strip().split('\n')
  pairs = [line.split('\t') for line in  lines]
  return pairs

# clean a list of lines
def clean_pairs(lines):
  cleaned = list()
  # prepare regex for char filtering
  re_print = re.compile('[^%s]' % re.escape(string.printable))
  # prepare translation table for removing punctuation
  #table = str.maketrans('', '', string.punctuation)
  for pair in lines:
    clean_pair = list()
    for line in pair:

      line = line.split()

      # remove punctuation from each token
      #line = [word.translate(table) for word in line]

      # store as string
      clean_pair.append(' '.join(line))
    cleaned.append(clean_pair)
  return array(cleaned)

# save a list of clean sentences to file
def save_clean_data(sentences, filename):
  dump(sentences, open(filename, 'wb'))
  print('Saved: %s' % filename)

# load dataset
filename = '/content/ben.txt'
```

```
doc = load_doc(filename)
# split into english-bengali pairs
pairs = to_pairs(doc)
# clean sentences
clean_pairs = clean_pairs(pairs)
# save clean pairs to file
save_clean_data(clean_pairs, 'english-bangla.pkl')
# spot check
for i in range(100):
  print('[%s] => [%s]' % (clean_pairs[i,0], clean_pairs[i,1]))
```

For non-latin scripts

```
import string
import re
from pickle import dump
from unicodedata import normalize
from numpy import array

# load doc into memory
def load_doc(filename):
  # open the file as read only
  file = open(filename, mode='rt', encoding='utf-8')
  # read all text
  text = file.read()
  # close the file
  file.close()
  return text

# split a loaded document into sentences
def to_pairs(doc):
  lines = doc.strip().split('\n')
  pairs = [line.split('\t') for line in  lines]
  return pairs
```

```
# clean a list of lines
def clean_pairs(lines):
  cleaned = list()
  # prepare regex for char filtering
  #re_print = re.compile('[^%s]' % re.escape(string.printable))
  # prepare translation table for removing punctuation
  table = str.maketrans('', '', string.punctuation)
  for pair in lines:
    clean_pair = list()
    for line in pair:
      # normalize unicode characters
      #line = normalize('NFD', line).encode('ascii', 'ignore')
      #line = line.decode('UTF-8')
      # tokenize on white space
      line = line.split()
      # convert to lowercase
      #line = [word.lower() for word in line]
      # remove punctuation from each token
      line = [word.translate(table) for word in line]
      # remove non-printable chars form each token
      #line = [re_print.sub('', w) for w in line]
      # remove tokens with numbers in them
    # line = [word for word in line if word.isalpha()]
      # store as string
      clean_pair.append(' '.join(line))
    cleaned.append(clean_pair)
  return array(cleaned)

# save a list of clean sentences to file
def save_clean_data(sentences, filename):
  dump(sentences, open(filename, 'wb'))
  print('Saved: %s' % filename)
```

```
# load dataset
filename = 'nds.txt'
doc = load_doc(filename)
# split into english-german pairs
pairs = to_pairs(doc)
# clean sentences
clean_pairs = clean_pairs(pairs)
# save clean pairs to file
save_clean_data(clean_pairs, 'english-LowSaxon.pkl')
# spot check
for i in range(100):
  print('[%s] => [%s]' % (clean_pairs[i,0], clean_pairs[i,1]))
```

A-2: Split Text

```
from pickle import load
from pickle import dump
from numpy.random import rand
from numpy.random import shuffle

# load a clean dataset
def load_clean_sentences(filename):
  return load(open(filename, 'rb'))

# save a list of clean sentences to file
def save_clean_data(sentences, filename):
  dump(sentences, open(filename, 'wb'))
  print('Saved: %s' % filename)

# load dataset
raw_dataset = load_clean_sentences('english-bangla.pkl')
```

```
# reduce dataset size
n_sentences = 4332
dataset = raw_dataset[:n_sentences, :]
# random shuffle
shuffle(dataset)
# split into train/test
train, test = dataset[:3332], dataset[1000:]
# save
save_clean_data(dataset, 'english-bangla-both.pkl')
save_clean_data(train, 'english-bangla-train.pkl')
save_clean_data(test, 'english-bangla-test.pkl')
```

A-3: Train Neural Translation Model

```
from pickle import load
from numpy import array
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
from keras.utils.vis_utils import plot_model
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import Embedding
from keras.layers import RepeatVector
from keras.layers import TimeDistributed
from keras.callbacks import ModelCheckpoint

# load a clean dataset
def load_clean_sentences(filename):
  return load(open(filename, 'rb'))

# fit a tokenizer
```

```
def create_tokenizer(lines):
  tokenizer = Tokenizer()
  tokenizer.fit_on_texts(lines)
  return tokenizer

# max sentence length
def max_length(lines):
  return max(len(line.split()) for line in lines)

# encode and pad sequences
def encode_sequences(tokenizer, length, lines):
  # integer encode sequences
  X = tokenizer.texts_to_sequences(lines)
  # pad sequences with 0 values
  X = pad_sequences(X, maxlen=length, padding='post')
  return X

# one hot encode target sequence
def encode_output(sequences, vocab_size):
  ylist = list()
  for sequence in sequences:
    encoded = to_categorical(sequence, num_classes=vocab_size)
    ylist.append(encoded)
  y = array(ylist)
  y = y.reshape(sequences.shape[0], sequences.shape[1], vocab_size)
  return y

# define NMT model
def define_model(src_vocab, tar_vocab, src_timesteps, tar_timesteps, n_units):
  model = Sequential()
  model.add(Embedding(src_vocab, n_units, input_length=src_timesteps, mask_zero=Tr
ue))
  model.add(LSTM(n_units))
```

```
  model.add(RepeatVector(tar_timesteps))
  model.add(LSTM(n_units, return_sequences=True))
  model.add(TimeDistributed(Dense(tar_vocab, activation='softmax')))
  return model

# load datasets
dataset = load_clean_sentences('english-bangla-both.pkl')
train = load_clean_sentences('english-bangla-train.pkl')
test = load_clean_sentences('english-bangla-test.pkl')

# prepare english tokenizer
eng_tokenizer = create_tokenizer(dataset[:,0])
eng_vocab_size = len(eng_tokenizer.word_index) + 1
eng_length = max_length(dataset[:, 0])
print('English Vocabulary Size: %d' % eng_vocab_size)
print('English Max Length: %d' % (eng_length))
# prepare  bangla tokenizer
ger_tokenizer = create_tokenizer(dataset[:, 1])
ger_vocab_size = len(ger_tokenizer.word_index) + 1
ger_length = max_length(dataset[:, 1])
print('Bangla Vocabulary Size: %d' % ger_vocab_size)
print('Bangla Max Length: %d' % (ger_length))

# prepare training data
trainX = encode_sequences(ger_tokenizer, ger_length, train[:, 1])
trainY = encode_sequences(eng_tokenizer, eng_length, train[:, 0])
trainY = encode_output(trainY, eng_vocab_size)
# prepare validation data
testX = encode_sequences(ger_tokenizer, ger_length, test[:, 1])
testY = encode_sequences(eng_tokenizer, eng_length, test[:, 0])
testY = encode_output(testY, eng_vocab_size)

# define model
```

```
model = define_model(ger_vocab_size, eng_vocab_size, ger_length, eng_length, 256)
model.compile(optimizer='adam', loss='categorical_crossentropy')
# summarize defined model
print(model.summary())
plot_model(model, to_file='model.png', show_shapes=True)
# fit model
filename = 'model.h5'
checkpoint = ModelCheckpoint(filename, monitor='val_loss', verbose=1, save_best_onl
y=True, mode='min')
model.fit(trainX, trainY, epochs=100, batch_size=64, validation_data=(testX, testY), ca
llbacks=[checkpoint], verbose=2)
```

A-4: Evaluate Neural Translation Model

```
from pickle import load
from numpy import array
from numpy import argmax
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import load_model
from nltk.translate.bleu_score import corpus_bleu

# load a clean dataset
def load_clean_sentences(filename):
  return load(open(filename, 'rb'))

# fit a tokenizer
def create_tokenizer(lines):
  tokenizer = Tokenizer()
  tokenizer.fit_on_texts(lines)
  return tokenizer

# max sentence length
```

```
def max_length(lines):
  return max(len(line.split()) for line in lines)

# encode and pad sequences
def encode_sequences(tokenizer, length, lines):
  # integer encode sequences
  X = tokenizer.texts_to_sequences(lines)
  # pad sequences with 0 values
  X = pad_sequences(X, maxlen=length, padding='post')
  return X

# map an integer to a word
def word_for_id(integer, tokenizer):
  for word, index in tokenizer.word_index.items():
    if index == integer:
      return word
  return None

# generate target given source sequence
def predict_sequence(model, tokenizer, source):
  prediction = model.predict(source, verbose=0)[0]
  integers = [argmax(vector) for vector in prediction]
  target = list()
  for i in integers:
    word = word_for_id(i, tokenizer)
    if word is None:
      break
    target.append(word)
  return ' '.join(target)

# evaluate the skill of the model
def evaluate_model(model, tokenizer, sources, raw_dataset):
  actual, predicted = list(), list()
```

```
  for i, source in enumerate(sources):
    # translate encoded source text
    source = source.reshape((1, source.shape[0]))
    translation = predict_sequence(model, eng_tokenizer, source)
    raw_target, raw_src,train = raw_dataset[i]
    if i < 10:
      print('src=[%s], target=[%s], predicted=[%s]' % (raw_src, raw_target, translation))
  actual.append([raw_target.split()])
  predicted.append(translation.split())
  # calculate BLEU score
  print('BLEU-1: %f' % corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
  print('BLEU-2: %f' % corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0)))
  print('BLEU-3: %f' % corpus_bleu(actual, predicted, weights=(0.3, 0.3, 0.3, 0)))
  print('BLEU4: %f' % corpus_bleu(actual, predicted, weights=(0.25, 0.25, 0.25, 0.25)))

# load datasets
dataset = load_clean_sentences('english-bangla-both.pkl')
train = load_clean_sentences('english-bangla-train.pkl')
test = load_clean_sentences('english-bangla-test.pkl')
# prepare english tokenizer
eng_tokenizer = create_tokenizer(dataset[:, 0])
eng_vocab_size = len(eng_tokenizer.word_index) + 1
eng_length = max_length(dataset[:, 0])
# prepare Bengali tokenizer
ger_tokenizer = create_tokenizer(dataset[:, 1])
ger_vocab_size = len(ger_tokenizer.word_index) + 1
ger_length = max_length(dataset[:, 1])
# prepare data
trainX = encode_sequences(ger_tokenizer, ger_length, train[:, 1])
testX = encode_sequences(ger_tokenizer, ger_length, test[:, 1])

# load model
model = load_model('model.h5')
```

```
# test on some training sequences
print('train')
evaluate_model(model, eng_tokenizer, trainX, train)
# test on some test sequences
print('test')
evaluate_model(model, eng_tokenizer, testX, test)
```