National College of Ireland

Neural Machine Translator aided translation of the English language to under-resourced language

MSc Research Project
Data Analytics

Rajdeep Rajan Karpe
Student ID: x17164851

School of Computing
National College of Ireland

Supervisor: Prof. Noel Cosgrave

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rajdeep Rajan Karpe
Student ID:	x17164851
Programme:	Data Analytics
Year:	2018
Module:	MSc Research Project
Supervisor:	Prof. Noel Cosgrave
Submission Due Date:	12/12/2019
Project Title:	Neural Machine Translator aided translation of the English language to under resourced language
Word Count:	7104
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th December 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Neural Machine Translator aided translation of the English language to under-resourced language

Rajdeep Rajan Karpe
x17164851

Abstract

Natural language Processing is the field of training the machine to learn human languages. Globally there are more than 7000 spoken languages. Only a few of the languages are considered as high resource languages. It has been observed that natural language processing techniques focus on high resource languages that have text corpora of millions of words. In this work, we try to develop the Neural Machine translator for low resource language where the Marathi language will be considered as low resource language, which has limited annotated corpora. Long short-term memory (LSTM) neural network with the word embedding technique will be used to translate English sentences to Marathi. Bilingual Evaluation Understudy (BLEU) score can be used as a validation score for predicted Marathi and English sentences. An accuracy of 78.59% is achieved after running 250 epochs on training data and for the validation of the same a BLEU score of 0.83 is attained by test data. This research promises to contribute to the development of under-resourced language translation techniques.

Keywords :*Neural Machine Translation, Natural Language Processing, Long Term Short Memory, Bilingual Evaluation Understudy, Encoder, Decoder, Epochs, Sequence to Sequence Modeling*

1 Introduction

Linguist resources, as defined by Schwenk and Douze (2017) are the assets of a language that can be used for communication, research, analysis, and translation. Such resources play an important part in the language learning process too. The study of a language purely depends upon its linguist resources and their number of multiple applications. These not only include the words, phrases, and character of the language, but they also include a deeper look of text fonts, textbooks, corpora, ontology, dictionary and more. On account of linguist resources, we can divide the whole resource library into three different levels. First, are the enabling resources that include basic input-output and representation methods? Second, are the language resources that refer to phrasal and characteristic entities of the language? Such entities are a part of lexicons, corpora, and thesauri of the language. Language resources are further divided into two types: online resources and offline resources. The online language resources are the one that can be used easily to learn the language explicitly. Most of such assets include basic words and characters of the language. But offline resources like exact pronunciations and spell tricks can be learned through offline language resources. Offline resources also include detailed meaning, synonyms, antonyms, grammatical parts of speech, guidance uses and more.

The third type of linguist resources is the recognizers, spell checker, translators, search engines, word predictors and more. Such tools help to make language integration in software, making it easier for users to work conveniently in their native languages.

Language has been a source for communication and understanding, not only between humans but also between machines Choukri (2019). The machine-readable sets of language data and assets that help in transformation and switching between multiple languages are known as language resources. The natural language speech and asset identification algorithms play a part in the extent of determining the versatility and expandability of the language. Language resources are written in various forms and techniques. These might include spoken corpora, computational lexica, terminology databases, speech collection, etc. all of these methods include defined and detailed algorithm which determines a specific manipulation of language assets for the processing and understanding. The machines are made capable of identifying these languages as soon as the machine identifies patterns of language assets defined in these algorithms. Machines also match patterns of multiple languages in a similar structure to translate and switch between the languages. Resourceful languages have hundreds of thousands of asset corpora mentioned in it. There are unlimited sentence structures formed with iterative processing of these asset corpora for the language. The machine identifies these corpora to generate signals for the understanding, acquisition, management, and customization of the language. The language recognition tools and software are an important part of the machine's capability to understand language resources and assets.

Language resources play an important part in the development and evaluation of machine language matters. The development of software which is capable of generating language resource patterns using specified algorithms from the provided language assets. Whereas evaluation includes detailed processes of searching, sorting and manipulating in language asset patterns. If a company develops the language recognition and switching feature in their product, they can gain a suitably high number of customers as more customers are attracted towards the application due to language proficiency. Thus, it becomes easier for customers to understand company policies. Although creating language proficiency is not cheap, but the customers and leads obtained after adding language integration to the software makes more profit, making the project more cost-effective. Thus, the circulation of money is obtained. Languages that can be identified through the brief definition of being resourceful or under-resourced. This definition is specified on the basis of the number of language resources and assets in the language library.

There are many languages that are full of resources, but their patterns produced are overlapped and mingled. This makes understanding difficult as the complex processes of language learning are mixed up. The evaluation software feels it difficult to sort and differentiate between two strings and patterns, which look very similar. Thus, such languages are declared as having fewer assets, as the provided assets aren't sufficient enough to enable proper machine-based understanding. Under-resourced languages are the languages that have been declared low in asset count, as the assets are not enough to be counted as appropriate for machines to conduct the language learning operations in the language. Such languages are not easily identified in written/typed forms as they don't possess fixed structures and identification abilities due to fewer assets for the machine to learn language synchronization. An under-resourced language is rarely found in interna-

tional software, as an option of language choice. This is because of the same reason of fewer assets.

EL-Haj et al. (2014) callout that such language is processed and taken through brief methods of asset generation so that they can include a different entity for each asset and the patterns generated with the available corpora are unique. Unique corporal patterns generated are which are successfully identifies with the machine learning algorithm. The under-resourced language sets which are successfully identified after the processing in machine learning compatibility algorithm are made capable of supporting and retrieving the data from the language library. They also presented the idea and methods that can be used effectively to handle the under-resourced languages in the processes of language manipulations. The major work of learning algorithms towards language processing is seen in the form of algorithms and methods that work especially to increase language resources of under-resourced languages and also help to evaluate their word count. One of such algorithms are LSTMs proposed by Barone (2016). Marathi is an Indo-Aryan language that has a lot of, more than 71 million speakers in Indian states of Maharashtra and surrounding, and also in some parts of Mauritius and Israel. It can be traced back to the 10th century when a dialect of Marathi evolved out from Sanskrit. Later, in the 11th century, people started to adopt alphabets and words to write the Marathi language . The Indian film industry also works efficiently for the production and release of films and movies in the Marathi language. Although it has a large number of speakers in the world, the Marathi language is considered as under-resourced language. People can know and learn the Marathi language if they are in contact with any native Marathi speaker. But learning this language from websites, seeking translation to other languages, and multilingual conversations can be difficult.

This paper discusses a deep learning approach using LSTMs with sequence to sequence model, to generate translators for the Marathi language. The experiment has been performed for different epochs, where we have achieved a quite well accuracy for training the data over LSTM. The validation of the model is done using the BLEU score. The paper also discusses the future possibilities of improving the model.

1.1 Research Objective

The important step for a researcher is identification of objective for study of a problem in a right direction. Objective of this research is as follows :

- To develop a neural machine translation for the low resources languages.
- To achieve higher accuracy for Marathi-English Translator.

1.2 Research Questions

- How accurately can a neural machine-based translator perform a word-level translation between the English language and the under-resourced language Marathi?

1.3 Scope of Investigation

This project is aimed to develop a Neural Machine translator for the Marathi language. These LSTM model are sequential models uses Encoder-decoder architecture to perform translation. The dataset has been taken from `http://www.manythings.org/anki/mar-eng.zip` which contains more than 36000 sentences of Marathi and English languages. Word Embedding techniques will be used to convert words into vectors. LSTM is a popular technique for neural machine translation which will be explored in our experiment and results can be analyzed.

2 Related Work

Gu et al. (2018a) said that Neural Machine Translation (NMT) could be integrated into under-resourced languages if their assets are manipulated and increased. He identified the idea of gathering more assets for such languages with machine learning. As a point of notion, the NMT is not only significant for the translation between high-resource languages. But, the systems that can translate to low-resourced languages also increase the number of uses for the same language. Therefore, NMT for under-resourced languages helps to gain more users. Also, people native to under-resourced language can be accessed by communicating to them in their language. Businesses can progress with this efficient technique. He focused on the creation of resources with robust MT technologies. The use of Bayesian models for language learning on machine systems is a good tool to make the languages able to be a part of NMT. He worked with his team to design methods and techniques to gather assets from low-resourced languages like data transfer and augmentation. Such techniques can assist in extracting hidden data assets in the language library. They also worked on rapid, deep learning and used this technique to create intelligent machine learning systems that can iterate the data mining processes in learning the under-resourced language. Thus, the system succeeded in gathering data from the under-resourced languages and develop an NMT for such systems. These NMT systems were not only capable of gathering data from under-resourced languages but also handling the high-resourced languages. All it required was a detailed dataset of the language resources from the assets library. The system would use effective machine learning technologies and methods to learn from the language library. The system also deployed Robust MT and rapid, deep learning algorithms to learn from the data from under-resourced languages. In this way, the NML for every type of language was developed. This system was intended to be used in the British territory to reduce the language conflicts between the regions and enhance international trade and communal affairs. This system was a clear use of AI technologies and their types.

Zoph et al. (2016) worked on similar strategies of NMT and emphasized the idea that the encoder-decoder framework is good to be used in high-resource languages for data processing. This framework uses a data matching and augmentation algorithm to memorize the data in a machine learning system. But, this method can't be implemented in low-resourced languages. Therefore, a transfer learning method is proposed which can operate in the machine learning process of under-resourced languages. This method works in two steps. The first step involves the extraction of similar components in a high-resourced and low resourced language. This extraction is done so that the data in high –resourced language can be used as a data source in the under-resourced lan-

guage. Thus, NMT can work on the under-resourced language with a comparatively higher number of resources. With this method, it was observed that the resulting NMT was more efficient and accurate. The extracted data sources were added in parallel to the original data source. Thus, the machine identified both sources as a part of a single language and doesn't identify the origin of the extracted data. Thus, NMT was improved.

EL-Haj et al. (2014) worked with the under-resourced Arabic language. They claim that many different ways can be used to illustrate the under-resourced languages that are groomed through algorithms. The first method is known as crowdsourcing, which helps to provide a rapid and cheap set of program instructions, which helps in fast language transformation and asset generation — next, transformation through a gold standard set, which is a benchmark of basic data generation. Primary language assets are transformed easily and rapidly through the gold standard sets. These are the algorithms that were involved in the naming and identification of primary language entities. Later, the same entities are derived, cut and merged with the older language assets. Thus, new data entities are made for the language asset library. Last and the most complicated method are to use the human effort to define each entity separately, and new details added manually in the system. The machine uses these words to learn language assets and use them for language proficiency and integration in the application. As a fact, the third method of the annual addition of the language entities is the most trusted and confirmed way of expanding datasets. Human has been closest to the languages. Native people, who own a particular language for centuries are more close and accurate about their language resources. As a fact, the languages keep going through changes and alterations. The native people can identify what the latest language resources are, but a machine won't. There is a great language asset available from the historical content of the languages but to differentiate between the latest and deed language assets. The native people understand the clear pronunciation and structure of each entity of the language. Therefore, the machines can never succeed to match the human capabilities for language resource recognition and expansion. Therefore, in various methods of language resource expansion, the third method is often used as a test case and a verification method to check the capabilities of the machine and the software included in the system. These methods also help to identify increase the language assets for the language by comparing it with the asset library of another high-resourced language. In this way, the asset comparison leads to the generation of similar assets. These assets are used for translators and dictionaries; by present these assets as literal meanings of each other.

Hasegawa-Johnson et al. (2017) mention that under-resourced languages may provide written texts and audio speeches. But, proper phonetics that can be used for ASR (Automatic Speech Recognition) is not available. ASR is a machine-learning application that can be used to recognize speech in any language, not told to the system before. But the language must be learned by the system to recognize the language. Some ways can be used to produce phonetic waveforms based on existing speech audios for ASR of under-resourced languages. Such techniques use probabilistic linguist transcripts that can help to produce waveforms of the required type. There are three sources of probabilistic transcripts used in this model. First, the self-learning which helps the ASR to identify the unknown speech and record it. This speech is then learned and used for speech recognition of the under-resourced language. This technique requires human involvement to train the ASR. Next, the crowdsourcing technique can be used to attain the machine

to identify unrecognized accents for the language. In this method, non-native people are made to hear a word and write it down. The resulting words are memorized by the machine making it capable of identifying different accents of people. Third, EEG which is a new type of coding is used. This method works in the electrocortical signals produced by the non-native listeners of the language when they hear the words of this language. As a result, enhance probabilistic data is recorded. In this way, an ASR of an under-resourced language could be made without taking help from the native speakers. The second and third methods were the most productive in this regard. EEG and crowdsourcing were used in combination, so any errors during the detection were also handled successfully. The team produces ASR for four different under-resourced languages without any assistance from natives.

Dereza (2019) has talked and researched the process of lemmatization, which is a simple method of treating a group of words (phrases or sentences) as a single item. A single item is comprised of various data packets identified by the dictionary distinctively. Some languages are structured in such a way that phrases and sentences comprise of various words, making lemmatization easier. But for other languages like Russian and Irish, the data lemmatization becomes difficult due to the morphology in the sentences. The process is difficult, yet, it can be completed. But for historical and low-resourced languages, the process is very difficult as packing a few assets in data sets leads to a further reduction in the asset count. This deficiency of data can be overcome by the process of sequence-to-sequence learning. This method is a feasible way to tackle the under-resourced languages during language processing processes. It was seen that 83,155 different sample words were handled by the sequence-to-sequence language iteration algorithm. The results were astonishing as the algorithm was able to produce 34,000 iterations. There was a high accuracy traced for the known vocabulary, which was 99.2%. Whereas, 64.9% of the unknown words were found to be accurate. The sequence-to-sequence model operates on the baseline model proposed by Dereza (2019)and the rule-based solution given by Segalovich (2003). Sailor et al. (2018) worked on Gujrati, an under-resourced Indian language. They tried to develop an efficient ASR for the assistance of illiterate farmers in the region of Gujrat, India. They used three algorithms for their ASR development: language modeling, phonetic modeling and featured learning. They especially worked to collect language entities used by farmers and in the agricultural sector to create a more helpful ASR. They used TDNN (Time delay neural network) on the methodology of Waibel, Hanazawa, Hinton, Shikano and Lang (1989) to produce phonetic modelling of the words in the list and used ConvRBM (convolution restricted Boltzmann Machine) proposed by Norouzi et al. (2009) and TEO (Teager Energy Operator) given by Beyramienanlou (2018) for the featured modelling of the Gujrati language assets. Later the RNN (Recurrent Neural networks) algorithm was used to perform the final language modeling. Thus, a good language library was established for the ASR machine learning algorithm to learn the library and identified that the NMT or (neural machine translation) is the most popular machine learning application. This algorithm is being used to create translations for multiple languages based on machine learning algorithms. A machine embedded with a machine learning algorithm is made to learn from the dataset of the language resources of the system. The machine learns the data and uses it to analyze and compare the assets with other languages. Thus, it makes a good tool for translation. But under-resourced languages face difficulty in translation because the few resources of the language make the training of the machine learning system difficult. The machine doesn't find a suffi-

cient number of language resources for the machine and faces difficulty in manipulating and learning from the data. This problem is the point of consideration in this report. We well are using machine learning and deep learning algorithms to produce NMT for under-resourced languages.

Grossfeld (2018) has presented the theories and concepts of deep learning and machine learning simply. All of the innovations and vast technologies included in the AI (artificial intelligence) can be packed into two distinct classes: machine learning and deep learning. Although, these two sound very similar but they have a great difference amongst them. It is important to know the difference between the two technologies to enable the most efficient learning. As a fact, machine learning is a technique that uses a certain algorithm to manipulate the data and learn certain attributes or data from it. It then applies those learned attributes to another system; it uses them to make decisions. Machine learning learns from a minute dataset and applies it somewhere. Thus, the produced outcomes are also learned by the machine and reused. Thus, the assets are increased gradually. On the other hand, deep learning can be called as a subset of machine learning as most of the applications and algorithms of deep learning are similar to those of machine learning. But, we can't grade both at the same ground as deep learning has very different capabilities. In machine learning, the algorithm works progressively and learns the data gradually. But, a certain stage is achieved when the algorithm used produced an inaccurate and unidentified attribute. Thus, an engineer or an AI expert has to step in. We can say that the machine learning algorithm requires human supervision and guidance. On the contrary, a deep learning algorithm can determine these errors and incorrect entities on its own based on the data it has learned previously. Therefore, such systems are more intelligent than machine learning systems.

Selamat and Akosu (2016) proposed a different algorithm for the understanding of under-resourced languages by the Machine learning algorithms. There is a good way of generating language assets and enriching the library of the under-resourced library by the word-length algorithm. This is a lexicon-based algorithm which makes use of a small number of language assets for the machine learning algorithm. This algorithm takes the shortest time period for a machine-learning algorithm to identify and learn an under-resourced language. Therefore, it can be said that if used on run-time, this algorithm can improve the activity of the algorithm as it would enhance the operations of the software in which it has been embedded. This algorithm has been identified on the document and sentence level as it produces accurate results. Thus, this algorithm can be used in word prediction, ASR, and language recognition software made with machine learning. This is a solution to the problem identified by Pienaar and Snyman (2010). They highlighted the fact that due to spell errors and mistypes, the word processor might fail to identify the language being typed. Therefore, it becomes difficult for the typist to trace the checks and errors. Also, this also created a problem for the search engines to identify the document of a particular language. This problem was also solved by the spell-check language identification algorithm which helped not only to resolve the language spell issues but also in language identification. In continuation, Beeslay (1988) considers language identification as the most important tool in computational linguists. He called every effort in vain if the machine or software fails to identify the language or identifies it as another language. According to him, the high resourced languages are also under risk of identification. For instance, the software may consider British English as American

English and start pointing out errors, whereas the errors were correct as British English library. Therefore, the identification of the language is the first and most important tool. Beeslay also proposed a computer program that could identify the languages, provided the language was embedded in the system library.

Jimerson et al. (2018)) wrote in his report that very few languages had been encoded for internet use. Only the encoded languages are available for use on the interest websites and search engines. This implies that a large number of high-resourced and under-resourced language is still available for being encoded. According to Ethnologue (2019), there are about 7111 spoken languages in the world with 23 of them being the most popular and covering about 50% of the world population. Enhanced information security can only be achieved if the systems are capable of handling the highest number of languages possible. But, creating systems for all languages is not possible. Therefore, a ready algorithm plugin must be available that can learn any language in a short time. All that is provided is to provide a ready to use the dataset of language assets. Thus, the large number of under resource languages can be mechanized on the algorithms verified on a specific language. Gu et al. (2018b) also proposed a machine learning algorithm that can work on under-resourced languages. This algorithm doesn't require individual entities for learning. Rather, it can learn from words and sentences, i.e., it can handle the lexical entities of the language. This method doesn't require another machine-learning algorithm to increase the entities count of the asset library.

Yılmaz et al. (2016) proposed an ASR that was capable of handling the bilingual accents. As a fact, this system was developed for the Frisian language. Most of the speakers of this language are bilingual. And often mix up the Frisian and Dutch languages. During testing, it was found that the ASR was recognizing the accent of a Frisian speaker as Dutch but wasn't recognizing the word, even when the Dutch language was fully operable in the ASR. Therefore, a bilingual accent ASR system was created using bilingual DNN (deep neural networks) that could produce accurate results from the speakers of multiple languages. DNN is a robust recognizer which handles the phenomenon of bilingual accent recognition. This algorithm also included the code-switching speech algorithm proposed by Ahmed and Tan (2012). Dereza (2019) also worked on dialect recognition. The big corpora of two different dialects of a language with 70% similarities often create a mess in machine recognition. Therefore, robust featured learning for dialect learning was integrated, which was capable of identifying the dialect based on varied accents. This approach was verified with conversational and read corpus making it feasible to be used as a tool for language identification.

Pipiras et al. (2019) have also performed detailed work to generate ASR that can identify the language in the speech based on phonetics only. They have worked out with the encoder-decoder method have also worked with the attention mechanism. This system is specially made to work with under-resourced languages. Since ASR is very easy and commonly found for high-resource languages, attempts have been made to work for under-resourced languages. The developed system was efficient enough to perfectly identify 99.3% of speech models and successfully identified 99.2% phrases perfectly.

Jimerson et al. (2018) worked to develop a mobile ASR application for an under-resourced language. This could be difficult as the language had very few assets available for the

working of the software. Therefore, the team included many youngsters and researchers to work on asset collection for this language. This included a novel method that was capable of producing synthetically developed data for the language library. The main aim of this project was to revitalize the Seneca language. This project also contacted people working on Seneca language, to collect a high number of assets for the system. Making the ASR was complex as Seneca comprise of longer words. Many of these words have similar initial phonetics and different final phonetics. The traditional ASR algorithm included the identification of the initial phonetics for the recognition of words and phrases. But, Seneca was dealt with differently because of longer words. An augmentation algorithm was used to handle this issue. Also, the sequence-to-sequence algorithm for ASR didn't work because of the Seneca language involver high utterance ratio, which is identified as an error by this algorithm.

He et al. (2018) identified the issue of similar pronunciations of words in languages. In low-resource languages, it becomes difficult to identify the similar pronunciations for the ASR setups. A cross-lingual probability for the multi-tasking NML can help this phenomenon. This system also works efficiently with the high-resourced languages and provides knowledge for a better language in ASR. This tool is assistive in cross-language switching and translation. Kann, Bjerva, Augenstein, Plank, and Sogaard (2018) have worked on the POS (Parts of speech) of under-resourced languages that are not easily identified on NML due to phrasal recognition. Strong character-level identification is required. They worked with three popular methods: lemmatization, autoencoding, and random string encoding, on 34 different under-resourced languages. The results revealed that this setup was able to provide a POS tagging for these languages in NML.

Bollmann et al. (2018) have worked with the historical languages that are dead by now. Even many of the languages have some assets which were popular once but are not used these days. Such languages may also be introduced to machine learning for research work. They have proposed that a historical or dead language must be introduced in the form of 10 different datasets to the NLP to enable multi-tasking machine learning. This enables the system to develop a machine integrated version of the dead language with lost data with very few assets. Chowdhury et al. (2018) experimented on NML with an MNMT (multimodal neural machine translation). This used pairs of low resource languages to see the results. Two datasets, one synthetic, and one manual were created to be added as parallel corpora for machine learning. The results showed the resulting language version was better when the synthetic data was used. This is because of the error-free nature of the synthetic data. Therefore, the synthetic datasets of the language can be used for imaging purposes. By imaging, we mean the integration of the same dataset onto another machine learning algorithm for integration purposes. Thus, a synthetic dataset obtained after the processing of an under-resourced language on the machine can help make further versions of the data. This extraction of data from the machine learning algorithm during its processing is known as the multimodal capability of the machine learning algorithm. Therefore, the one-time effort is required to produce the synthetic data from any mean for an under-resourced language.

As mentioned by in his blog, machine learning is the simplest algorithm used to parse data and learn from it. This learning is not limited to learning, but the system can also predict and determine the attribute existing in the world based on data learned previ-

ously. Such systems are developed by training the system with a set of data. The system uses special algorithms to learn these items. Later, the system learns the "how-to" of the algorithm and does it for learning. Thus, the machine is capable of learning and growing its compatibility gradually. On the other hand, deep learning is considered as the base to start machine learning. The first running algorithm must be capable of identifying entities and verifying their correctness. In this way, deep learning can produce a set of appropriate data from very few asset counts. Thus, the machine learning algorithm can operate successfully later on. But, as time passes, the system wouldn't be able to identify the incorrect data entities produces. For the detection of incorrect data and verify the entities produced, a deep learning algorithm guards the machine learning algorithm. This is very important as the system would collapse as soon as the machine starts to learn incorrect data. The machine learning algorithm individually requires a human to supervise the data learned. The human has to fix what the machine has learned wrong. This can be hectic in case of a large number of data. Even if a small data entity is remained unchecked in the system, the machine would manipulate it and multiply, producing a large number of incorrect data entries. This can lead to failure of the system. Therefore, we need to have a guide over the machine learning algorithm. To avoid human involvement in the machine learning algorithm, a deep learning algorithm is the best tool to serve this task perfectly.

Barone (2016) has talked about different deep machine learning algorithms like LSTMs, Resents, GRUs, and others. All of these architectures and various others require pass-through connections. They have worked to increase the number of input lines in pass-through connections for each algorithm and observed the changes. It is also found that the number of more or fewer lines doesn't affect the working of the network. By this, we can extract the meanings that the number of language datasets is not restricted in any deep learning algorithm. The user can select the number of a deep learning algorithm as per their convenience. For NML, deep learning also plays a good role when the under-resourced languages are integrated on the machine; the machine learning algorithms aren't effective enough.

Schwenk and Douze (2017) have worked out to find whether machine learning can identify multiple languages within a single sentence or not. They tried to create a sentence entity with words form six different languages and used distinct deep learning frameworks to identify these. Algorithms were redesigned for the sentence level. Most of the deep learning algorithms worked on the single character level. Therefore the algorithms were modified and used for the sentences. Up to 1.4M sentences were compared, and different syntax and structures were followed. It was found that most of the deep-learning algorithm successfully recognized the embedded under-resourced languages in the sentences as they were added through synthetic data sets. The high-resource languages that were added manually by their datasets were felt with difficulty. Also, this hypothesis was experimented to check if the high-resource language with synthetic datasets can be efficient in this system or not. The results were positive. Therefore, we conclude that the data obtained due to synthetic extracted of data were more integrated by the machines. Synthetic data sets can be reused for other algorithms, too, enabling lesser time consumption. Also, NML and ASR can be done with ease.

3 Methodology

The key steps of our methodology are Data preprocessing, Sequence to Sequence model, Data training using LSTM and Testing. The proposed framework will be used for Marathi-English translation. In the following subsection, we will discuss an overview of Neural Machine Translation, sequence to sequence model and word embedding techniques. Further, we will investigate the LSTM model for training our data.

3.1 Neural Machine Translation (NMT)

Neural Machine Translation is a neural network-based technique to increase the fluency and accuracy of machine translation. The NMT is purely based on Encoder-decoder architecture. The RNN (Recurrent Neural Network) is the most common type of network used in neural machine translation. Recurrent Neural Network involves cyclic structure and evolved as a very effective method to solve complex sequences to sequence problems for a large amount of data. Sequence to Sequence models is used to extract the relationship between the pairs of two different languages. The encoder and decoder are the two important parts of the Seq2seq model. In our proposed work both encoder and decoder are LSTM models. Encoder collects input from the source and based on these encoding vectors decoder produces the output.

3.2 Data collection

The dataset of Marathi and English translation has been taken from the following website `http://www.manythings.org/anki/mar-eng.zip`. The dataset contains the 36,504 English sentences with their Marathi translations. Where the maximum word length for Marathi languages is 37 and English is 34.

3.3 Data Preprocessing

The common step before building the model is data preprocessing. For our experiment data preprocessing steps involve Data Cleaning and Data Preparation steps. In Data cleaning we lowercase all the English characters, remove quotes, special characters, extra spaces and numbers from the data. Start and end token to the target sequence have also been added. Whereas, Data preparation steps are the generation of the vocabulary of English and Marathi words, find out the maximum length of the source and target sequences and tokenize the words.

3.4 Encoder

In our experiment both the encoder and decoder typically are the LSTM models. Encoder collects the input sequences from the source and converts it in the vectors called as internal state vectors. The internal state vectors in the encoder are nothing but used to summarize the information generated from the source. Our proposed method used word-level neural machine translation so every word in the sentence will be considered as a step. Word embedding technique will be used to map each word with a fixed length of the vector. The internal state vector is the combination of hidden state and cell state(hi, ci). The output sequence of the encoder is discarded. The example of encoder is shown in fig 1

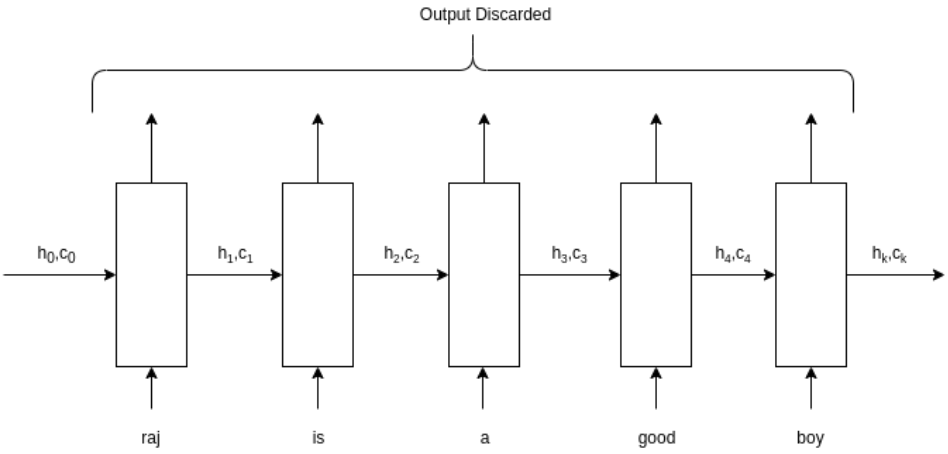

Figure 1: Encoder LSTM

3.5 Decoder

The final internal state vectors of the encoder are the initial state for the decoder. The data gets trained in the decoder using the LSTM model. In the case of the decoder, the output sequence is generated at every step. For efficiently training of RNN model we are using teacher forcing technique where the input at every step is the output of the previous step. The decoder example is shown in fig 2

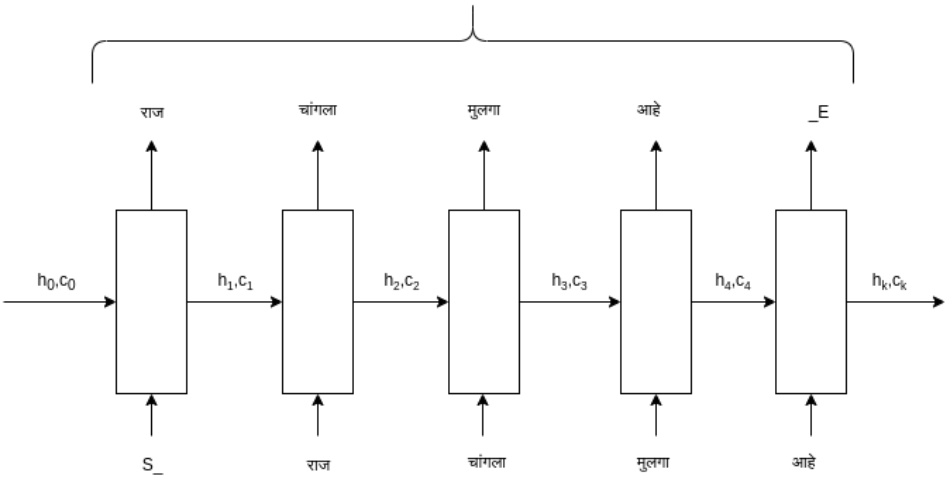

Figure 2: Decoder LSTM

4 Design Specification

This section defines the workflow of our proposed work which will be used to predict the output sequence. The detailed explanation of each section is described in the methodology section. The first step is to preprocess the data for which involves various states such as Data cleaning, remove numbers or digits, etc. To train the data sequence to the sequence model have been used where the Input sequence has been provided to Encoder followed

by decoder LSTM. To prevent the memory overhead, training data is provided in batches of 128 to LSTM for training the model effectively. The data has been trained for different epochs 50, 100 and 500. Where we noticed a significant improvement in the accuracy of the training model and a significant reduction in the loss. The flow diagram of proposed work is shown in Figure 3

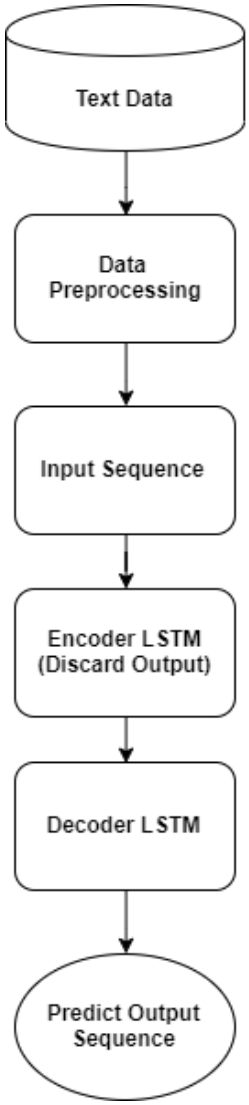

Figure 3: Flow Diagram of Proposed Work

5 Implementation

In order to implement the proposed work, the Graphics Processing Unit (GPU) is required. We have used Google colab service to train our model. Google colab provides 1xTesla K80 GPU, having 2496 CUDA cores with 12GB of GDDR5 VRAM. The time taken to train the model is 90 minutes for 50 epochs. To save the memory and train the model efficiently we have trained the data in batches with the batch size of 128. The algorithm used to train the model is Long short-term memory (LSTM), which has been used for both encoder and decoder. The output sequences of the encoder are discarded

in while preserving the internal states. The hidden state and cell state are the two states which combined make the internal state. The decoder takes these internal states as input and produces the output sequence at each step. The experiment will be performed over 50,100 and 250 number of epochs and to validate the model performance BLEU score have been used.

6 Evaluation

In this experiment, the model has been trained several times over the different epochs in order to predict better output sequences. The training accuracy and loss have been observed for different epochs. To validate the model, BLEU (Bilingual Evaluation Understudy) score has been used as an evaluation metric. The value of the BLEU score lies between 0 and 1.

6.1 Training of LSTM over 50 epochs

In the first experiment of our proposed work, we have used 50 epochs to train Long short term memory model. The dataset has been splitted into training and testing, where 90% of the data is used as training data and rest 10% is used as a testing data. We have also observed training accuracy and loss of the model for 50 epochs. Which is shown in fig 4 & 5

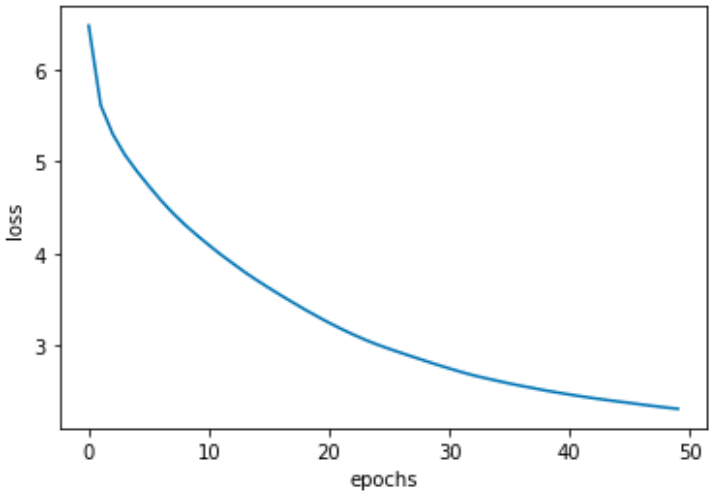

Figure 4: Epochs vs Loss

From the following graph 4 & 5 it has been observed that over every epoch loss is reducing and there is an increase in the accuracy of the model. The highest accuracy has been achieved by over 50 epochs is 64.62%. BLEU score between all the actual and predicted sentences is 0.429 over 50 epochs.

6.2 Training of LSTM over 100 epochs

The second experiment of our proposed work has been performed over 100 number of epochs, where the training model is Long short term memory (LSTM). The dataset has

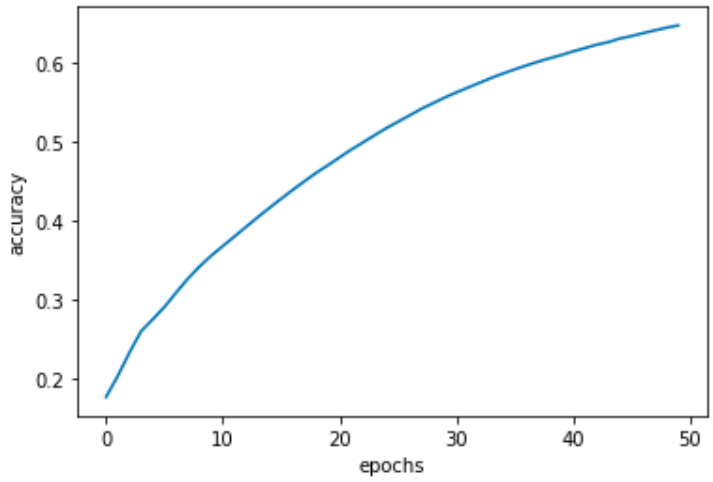

Figure 5: Epochs vs Accuracy

been split into training and testing set, where 90% of the data is used as training data and the rest 10% is used as a testing data. We have also observed training accuracy and loss of the model for 100 epochs. Graph is shown in fig 6 & 7

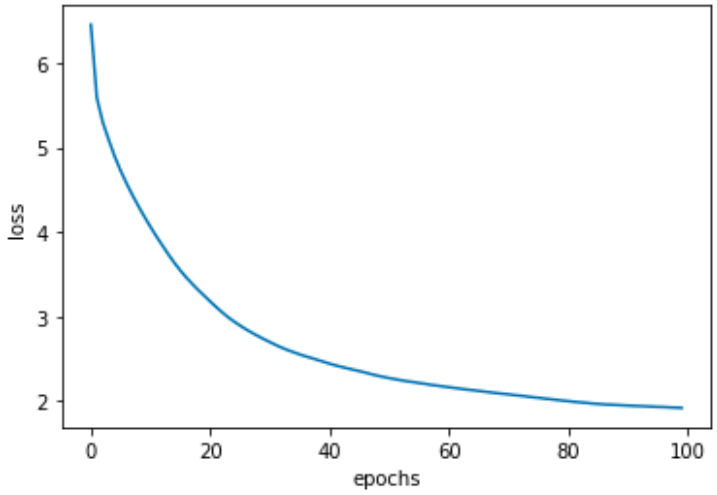

Figure 6: Epochs vs Loss

The highest accuracy achieved is 71.80% to train the LSTM model over 100 epochs. From Experiment 6.1 it has been observed that, as there is an increase in the number of epochs there is an increase in the accuracy and loss is reducing over every epoch. The BLEU score for all the actual and predicted sentence is 0.79.

6.3 Training of LSTM over 250 epochs

The final experiment of our proposed work has been performed over 250 number of epochs, where the training model is Long short term memory (LSTM). The dataset has been split into training and testing set, where 90% of the data is used as training data and the rest

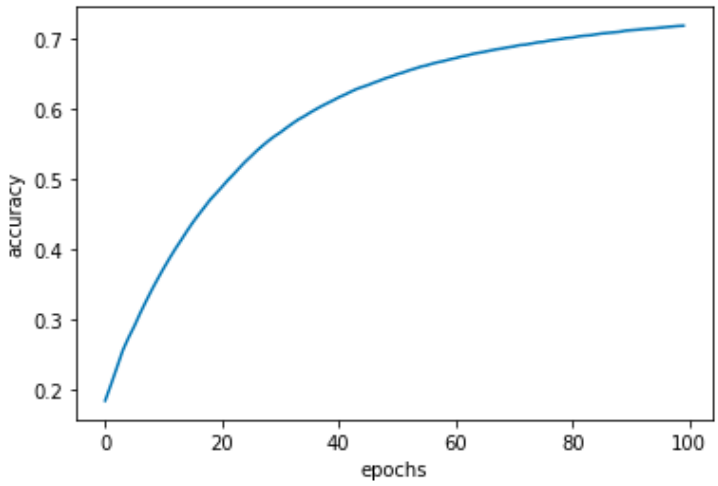

Figure 7: Epochs vs Accuracy

10% is used as a testing data. We have also observed training accuracy and loss of the model for 250 epochs. Graph is shown in fig 8 & 9

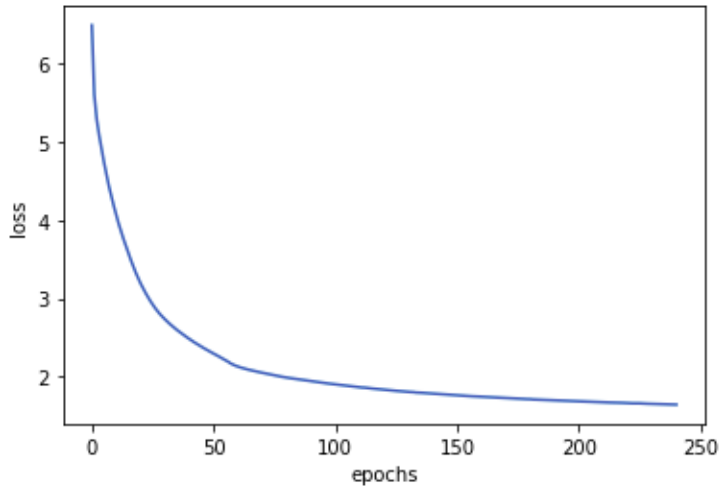

Figure 8: Epochs vs Loss

The highest accuracy achieved is 78.59% to train the LSTM model over 250 epochs. From Experiment 6.1 & 6.2 it has been observed that, as there is an increase in the number of epochs there is an increase in the accuracy and loss is reducing over every epoch. The BLEU score for all the actual and predicted sentence is 0.83.

6.4 Discussion

BLEU is a score for evaluating the quality of text which has been translated using a machine from one natural language to another language, BLEU score will be used as a validation score for our model. It has been also observed over the every there was an increase inaccuracy. we can also say that loss and accuracy are inversely proportional to

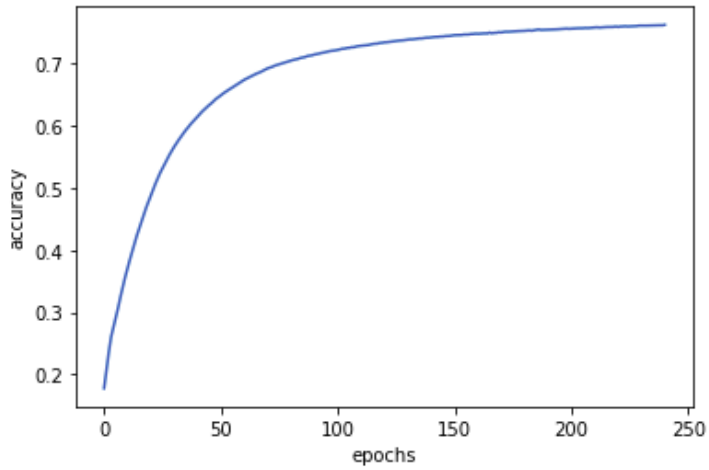

Figure 9: Epochs vs Accuracy

each other. As loss decreases the accuracy increases. The LSTM model requires a large amount of data but still, the performance of the model seems quite well.

7 Conclusion and Future Work

In this experiment, we have proposed a sequence to sequence model using LSTM to translate the English language into Marathi. Marathi is one of the low resource languages, which has limited annotated corpora. There are many of these languages, where sequence to sequence model using LSTM can be helpful. We have achieved an accuracy of 70% for training data. The BLEU score has been used to evaluate the performance of the model. This method is a very effective method for working on low resource languages. The limitation of this work is due to the availability of more data. The performance of the model can be improved by training the model with more data, where the hyper-parameters can be tuned such as batch size, learning rate, dropout rate, etc. A more complex model such as 'attention' can be explored to achieve effective results. An increase in the size of data requires more resources, which is another limitation of these methods.

References

Ahmed, B. H. A. and Tan, T. (2012). Automatic speech recognition of code switching speech using 1-best rescoring, *2012 International Conference on Asian Language Processing*, pp. 137–140.

Barone, A. V. M. (2016). Low-rank passthrough neural networks.

Bollmann, M., Søgaard, A. and Bingel, J. (2018). Multi-task learning for historical text normalization: Size matters.

Chowdhury, K. D., Hasanuzzaman, M. and Liu, Q. (2018). Multimodal neural machine translation for low-resource language pairs using synthetic data.

Dereza, O. (2019). Lemmatisation for under-resourced languages with sequence-to-sequence learning: A case of early irish, *in* G. Wohlgenannt, R. von Waldenfels, S. Toldova, E. Rakhilina, D. Paperno, O. Lyashevskaya, N. Loukachevitch, S. O. Kuznetsov, O. Kultepina, D. Ilvovsky, B. Galitsky, E. Artemova and E. Bolshakova (eds), *Proceedings of Third Workshop "Computational linguistics and language science"*, Vol. 4 of *EPiC Series in Language and Linguistics*, EasyChair, pp. 113–124.
URL: *https://easychair.org/publications/paper/Qv52*

EL-Haj, M., Kruschwitz, U. and Fox, C. (2014). Creating language resources for under-resourced languages : Methodologies, and experiments with arabic, *Language Resources and Evaluation* .

Gu, J., Hassan, H., Devlin, J. and Li, V. O. (2018a). Universal neural machine translation for extremely low resource languages, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, New Orleans, Louisiana, pp. 344–354.
URL: *https://www.aclweb.org/anthology/N18-1032*

Gu, J., Hassan, H., Devlin, J. and Li, V. O. K. (2018b). Universal neural machine translation for extremely low resource languages.

Hasegawa-Johnson, M. A., Jyothi, P., McCloy, D., Mirbagheri, M., d. Liberto, G. M., Das, A., Ekin, B., Liu, C., Manohar, V., Tang, H., Lalor, E. C., Chen, N. F., Hager, P., Kekona, T., Sloan, R. and Lee, A. K. C. (2017). Asr for under-resourced languages from probabilistic transcription, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **25**(1): 50–63.

He, D., Lim, B. P., Yang, X., Hasegawa-Johnson, M. and Chen, D. (2018). Improved asr for under-resourced languages through multi-task learning with acoustic landmarks.

Jimerson, R., Simha, K., Ptucha, R. and Prud'hommeaux, E. (2018). Improving asr output for endangered language documentation, *The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages* .
URL: *http://par.nsf.gov/biblio/10087204*

Norouzi, M., Ranjbar, M. and Mori, G. (2009). Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2735–2742.

Pipiras, L., Maskeliunas, R. and Damasevicius, R. (2019). Lithuanian speech recognition using purely phonetic deep learning, *Computers* **8**: 76.

Sailor, H., Patil, A. and Patil, H. (2018). Advances in Low Resource ASR: A Deep Learning Perspective, *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, pp. 15–19.
URL: *http://dx.doi.org/10.21437/SLTU.2018-4*

Schwenk, H. and Douze, M. (2017). Learning joint multilingual sentence representations with neural machine translation.

Segalovich, I. (2003). A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine., pp. 273–280.

Selamat, A. and Akosu, N. (2016). Word-length algorithm for language identification of under-resourced languages, *Journal of King Saud University - Computer and Information Sciences* **28**(4): 457 – 469.
URL: *http://www.sciencedirect.com/science/article/pii/S1319157815000609*

Yılmaz, E., van den Heuvel, H. and van Leeuwen, D. (2016). Investigating bilingual deep neural networks for automatic recognition of code-switching frisian speech, *Procedia Computer Science* **81**: 159 – 166. SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia.
URL: *http://www.sciencedirect.com/science/article/pii/S1877050916300588*

Zoph, B., Yuret, D., May, J. and Knight, K. (2016). Transfer learning for low-resource neural machine translation, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Austin, Texas, pp. 1568–1575.
URL: *https://www.aclweb.org/anthology/D16-1163*