NLP-Project : Group 18
Official Baatmi

Priyam Bajpai	Dishant Tayade	Satyam Kumar Singh	Raghav Garg	Happy Kumar
S20190010144	*S20190010175*	*S20190010158*	*S20190010148*	*S20190010061*
UG-3, CSE	*UG-3, CSE*	*UG-3, CSE*	*UG-3, CSE*	*UG-3, CSE*
priyam.b19@iiits.in	dishanthiraman.t19@iiits.in	satyamkumar.s19@iiits.in	raghav.g19@iiits.in	happy.k19@iiits.in

***Abstract*—The Indian Subcontinent is a combination of many nations which led to the incorporation of many languages, the major ones being the Indo-Aryan languages spoken by 78.05% of Indians and the Dravidian languages spoken by 19.64% of Indians. Both language families together are sometimes known as Indic languages. Taking into consideration the vast population of the nation and its language diversity, it is very difficult to have an exact count of the languages spoken here. Languages like Marathi, Tamil, Telugu and Bengali are one of the most spoken languages in the nation after Hindi and English. In the fast growing demand of data today, need of a model that inter-converts texts between these regional languages to Hindi/English has become quite obvious. In this paper, we aim to propose a model that uses various NLP pre-processing techniques and an LSTM model to inter-convert sentences between Hindi and Marathi language. Based on the parallel corpus of around 40,000 sentences as input, the model gives an accuracy of around 67% in the train data.**

***Index Terms*—NLP, POS, N-gram, seq2seq, LSTM, NMT**

I. Introduction

In this world, population increases daily. Due to this massive rise, people tend to retain their roots and customs, sometimes in form of religion, other times in form of language. But due to this, a plethora of languages coexist together and to solve this problem translation of these many languages becomes a pivotal task.

To make translation easier, Natural language processing (NLP) is done as it automatically converts one natural language into another with the help of Machine translation (MT),thus preserving the meaning of the input text, and producing fluent text in the output language. But translation is not as easy as it seems because many languages have different genealogical families i.e, their origins are widely different from each other.

One such example of genealogical language families is Indic languages which are made up of Indo-Aryan languages and Dravadian languages. Here in figure 1 language distribution across India is presented. But even with Indo-Aryan and Dravadian languages being of the same family, they differ widely.

According to the Indian Language policy, it states that the state government can run its daily work on three languages - Hindi, English or any other constitutionally recognised scheduled language. The state of Maharashtra is one such

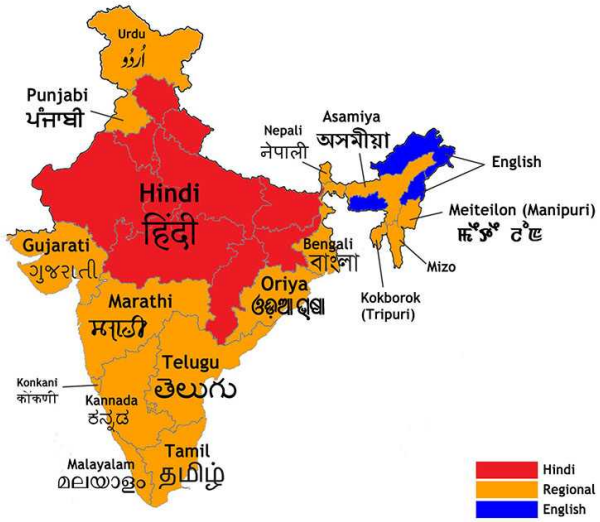

Fig. 1. Language distribution

state, that covers its officially written documents in its regional language Marathi.There is hardly any paperwork that is done in any other language other than Marathi. According to the Maharashtra Official Languages Act, 1964, it states that Marathi is the official language in Maharashtra, and should be used as per the three-language formula of the Centre, i.e, Marathi should be used along with Hindi and English. This circular also asked concerned offices to use Marathi along with English and Hindi languages in all written and oral examinations conducted during staff recruitment as Marathi is the *Rajbhasha* or the state language of Maharashtra.

As one can see Language translation is one of the most tedious tasks in a country as diverse as India but for sharing information, it becomes more necessary. To solve these problems, our model attempts Hindi to Marathi language conversion. There were many languages to choose from, but there are two reasons for taking up Marathi and Hindi.

- Both share the same script Devanagari which makes it easier to build and train the model.
- The group members have a fair knowledge of these two languages, which makes it easier to work on new data, and make our own models for POS Tagging and

Tokenization.

For our model we have planned to use a part of the Samanantar dataset which is probably one of the largest datasets available for training our model, and also use a seq2seq model which is a method of encoder-decoder based machine translation and language processing that maps an input of sequence to an output of sequence with a tag (for our model specially POS tagging) and attention value. The idea is to use 2 RNNs that will work together with a special token and try to predict the next state sequence from the previous sequence.

II. State of the Art/Background

Both the languages are members of the same family i.e. Indo-Aryan family of languages. They share a lot of features from script to phonemes and vocabulary. The problem statement was picked up from the Similar Language Task (SLT) from WMT 2020. Hindi-Marathi translation lacks background work.However, similar work is found on Hindi-Nepali pair at WMT19 shared task of similar language translation Laskar et al. (2019). The literature survey mainly focuses on NMT for low resource language pairs since NMT outperforms conventional SMT on low resource pairs like English to Mizo,English to Hindi, English to Punjabi, and English to Tamil Pathak et al. (2018); Pathak and Pakray (2018); Laskar et al. (2019). It is noticed that training performance improves while parallel training data increases. For low resource languages, it is difficult to collect parallel data unlike monolingual data which is easily found through online sources. Hence, monolingual based NMT systems are introduced to enhance the translation quality of low resource language pair translation. To get the advantage of monolingual data, unsupervised pre-train methods are introduced Ramachandran et al. (2017); Variš and Bojar (2019). Conneau and Lample (2019) proposed XLM based on bidirectional encoder representations from transformers (BERT) where the contextual language model is built with words based on preceding and succeeding context. No work has been done on Hindi-Marathi low resource language pair with such advanced NMT based approach, from the best of our knowledge. Our work investigates XLM model on Hindi-Marathi low resource language pair translation. NUIG-Panlingua-KMI submission to WMT 2020 seeks to push the state-of-the-art in Similar Language Translation Task for HindiMarathi language pair. As part of these efforts, we conducteda series of experiments to address the challenges for translation between similar languages. Among the 4 MT systems prepared under this task, 1 PBSMT systems were prepared for HindiMarathi each and 1 NMT systems were developed for HindiMarathi using Byte PairEn-coding (BPE) into subwords. The results show that different architectures NMT could be an effective method for developing MT systems for closely related languages. Our Hindi-Marathi NMT system was ranked 8th among the 14 teams that participated and our Marathi-Hindi NMT system was ranked 8th among the 11 teams participated for the task. Mujadia et.al. experimented with attention based recurrent neural network architecture (seq2seq) for this task. We explored the use of different linguistic features like POS and Morph along with back translation for Hindi-Marathi and Marathi-Hindi machine translation. In their paper, Amit et.al. use the Transformer networkbased NMT system (Vaswani et al., 2017) because it is among the state of the art models for machine translation. The work reported for this shared task is an extension of the work done by (Kumar and Singh, 2019) for similar languages task for 2019, which had also used a transformer based NMT system. Our work revolves around a similiar attempt for Indic language translation for hindi and marathi pair of corpora.

III. proposed system

The whole work is divided into four parts -

- Text Input, pre-processing and tokenization
- N-gram Model for both the languages
- POS Tagger for both the languages
- LSTM model for translation

We will look into details about each process and how was it implemented in the following sections.

A. *Text Input, pre-processing and tokenization*

*1) **Text Input**:* We have taken our train dataset from the Samanantar Dataset from AI4Bharat and used the Hindi-Marathi parallel corpora that contains approx 0.2 million sentences. We have utilised 40 thousand sentence, and pre-processed it to train our LSTM model on it.

*2) **Pre-processing**:* For pre-processing, we needed our own list of stop-words that cater to our need of the devanagari script. Also we required some method to remove hindi numerals used extensively in the dataset. We also tried out methods to mine and remove english words between the sentences in the corpora.

*3) **Tokenization**:* Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units called tokens, which is done after our data is cleaned with the help of previous two steps. These tokens help in understanding the meaning of our selected corpus by analyzing the sequence of the words and later processing our data for further analysis and also for developing the model for the NLP project.

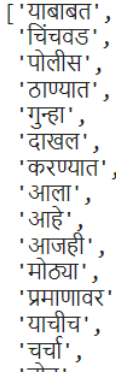

Fig. 2. Tokenization of marathi corpus

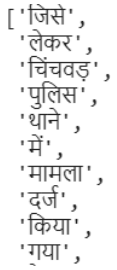

Fig. 3. Tokenization of hindi corpus

Tokenization is done in our code through the function word_tokenize and stored in the variable **finlist** which is shown in figure 2 and figure 3 for marathi and hindi corpus repectively.

B. N-gram

N-gram is a probabilistic model that's trained on a corpus of text to do a variety of tasks like speech recognition, machine translation and predictive text input. An N-gram model is one type of a Language Model (LM), which is about finding the probability distribution over word sequences.

For instance take these two sentences: "There was heavy rain" vs. "There was heavy flood". From experience, we know that the former sentence sounds better. An N-gram model will tell us that "heavy rain" occurs much more often than "heavy flood" in the training corpus. Thus, the first sentence is more probable and will be selected by the model.

A model that simply relies on how often a word occurs without looking at previous words is called unigram. If a model considers only the previous word to predict the current word, then it's called bigram. If two previous words are considered, then it's a trigram model.

An n-gram model for the above example would calculate the following probability:

$$P('There\ \ was\ \ heavy\ \ rain') = \tag{1}$$

$$P('There',' was',' heavy',' rain') \tag{2}$$

$$= P('There')P('was'|'There')P('heavy'|'Therewas') \tag{3}$$

$$P('rain'|'Therewasheavy') \tag{4}$$

Since it's impractical to calculate conditional probabilities of long sentences, one can use Markov assumption and approximate this to a bigram model:

$$P('There\ \ was\ \ heavy\ \ rain') \approx \tag{5}$$

$$P('There')P('was'|'There')P('heavy'|'was') \tag{6}$$

$$P('rain'|'heavy') \tag{7}$$

Seeing uni-gram and bi-gram, it came to our understanding that by calculating probabilities, we can calculate frequency of each and every word, better understand phrases that occur simultaneously and assigning specific meaning to these phrases. So we implemented an N-gram model, taking variable N to be the maximum length of any sentence present in both the datasets of hindi and marathi. Then this N-gram model counts how often word sequences occur in corpus text and then estimated the probabilities with which a word can appear in the translation from marathi to hindi.

By calculating the most frequents phrases via uni-gram and bi-gram, we create a dictionaries like

- n_gram_prob_mar for marathi corpus
- n_gram_prob_hi for hindi corpus

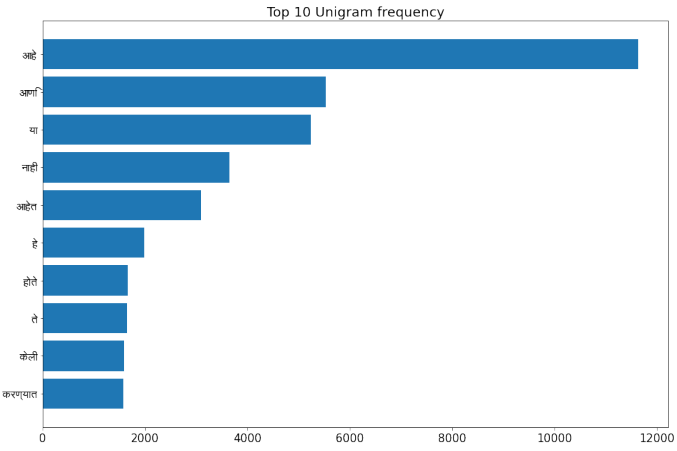

Fig. 4. Uni-gram for marathi corpus

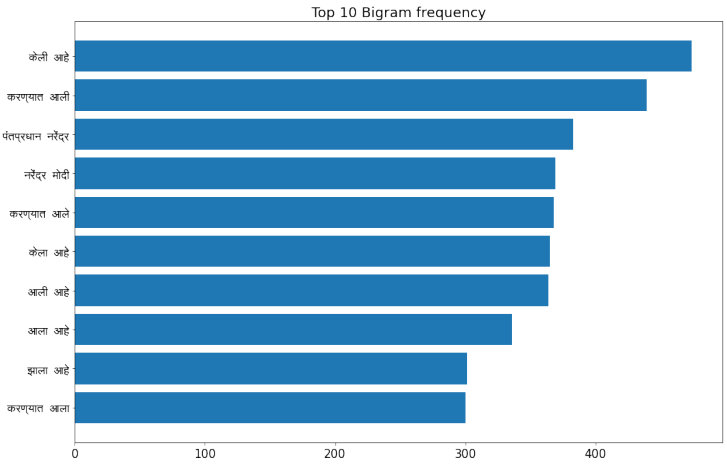

Fig. 5. Bi-gram for marathi corpus

We calculate frequency of uni-gram and bi-gram through consecutive loops.

Here in figure 4 and figure 5, it represents the uni-gram and bi-gram of Marathi corpus through the help of calc_ng_pro_mar function.

Here in figure 6 and figure 7, it represents the uni-gram and bi-gram of hindi corpus through the help of calc_ng_pro_hin function

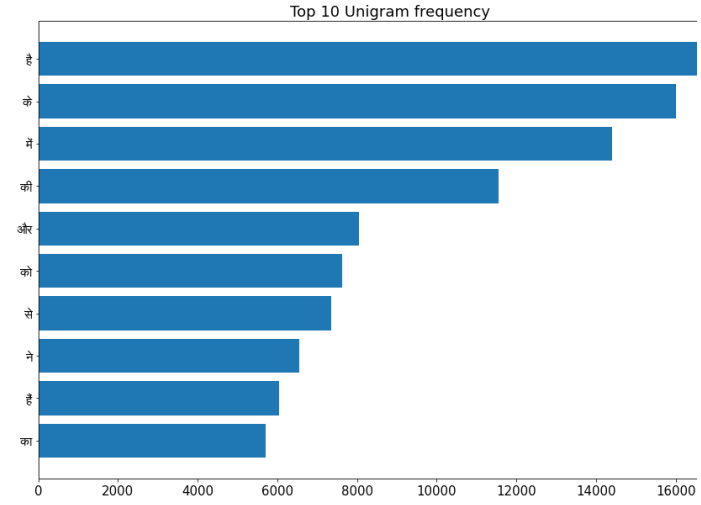

Fig. 6. Uni-gram for hindi corpus

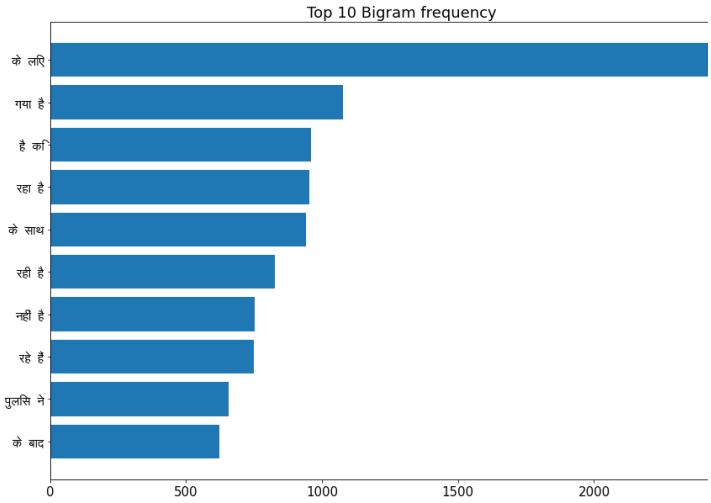

Fig. 7. Bi-gram for hindi corpus

C. *POS Tagging*

Part-of-speech (POS) tagging is a popular Natural Language Processing process which refers to categorizing words in a text (corpus) in correspondence with a particular part of speech, depending on the definition of the word and its context.

In corpus linguistics, Parts-Of-Speech or POS tagging, also called grammatical tagging is the process of marking up a word as nouns, verbs, adjective, adverb, etc. Once performed by hand, POS tagging is now done in the context of computational linguistics, using algorithms which associate discrete terms, as well as hidden parts of speech, by a set of descriptive tags. POS-tagging algorithms fall into three distinctive groups: rule-based, stochastic and hybrid.

Part-of-speech tagging is harder than just having a list of words and their parts of speech, because some words can represent more than one part of speech at different times, and because some parts of speech are complex or unspoken. This is not rare—in natural languages (as opposed to many artificial languages), a large percentage of word-forms are ambiguous. For example, even ”dogs”, which is usually thought of as just a plural noun, can also be a verb: The sailor dogs the hatch.

Correct grammatical tagging will reflect that ”dogs” is here used as a verb, not as the more common plural noun. Grammatical context is one way to determine this; semantic analysis can also be used to infer that ”sailor” and ”hatch” implicate ”dogs” as 1) in the nautical context and 2) an action applied to the object ”hatch” (in this context, ”dogs” is a nautical term meaning ”fastens (a watertight door) securely”). So, a Rule based POS tagger cannot give correct results always, so there is a need to consider the probability for each word and probablity of the word to be a specific part of speech. This is the reason, that motivates the use of Stochastic method for developing Parts of Speech tagger.

Steps in POS tagging

- Obtain a tagged data set for training.
- Using Hiddem Markov Model to identify Transmission and Emission Probabilities
- Apply Viterbi Algorithm on Testing Data Set.
- Output the tagged sequence with the highest probability.

We have obtained the source data set containing Marathi and Hindi Sentences with tagged POS from Marathi and Hindi sections of the universal dependencies corpus. The source corpora, documentation and credits can be found at http://universaldependencies.org

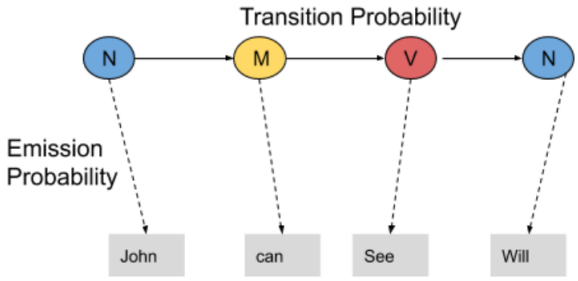

Fig. 8. Language distribution

Stochastic (Probabilistic) tagging: A stochastic approach includes frequency, probability or statistics. The simplest stochastic approach finds out the most frequently used tag for a specific word in the annotated training data and uses this information to tag that word in the unannotated text. But sometimes this approach comes up with sequences of tags for sentences that are not acceptable according to the grammar rules of a language. One such approach is to calculate the probabilities of various tag sequences that are possible for a sentence and assign the POS tags from the sequence with the highest probability. Hidden Markov Models (HMMs) are probabilistic approaches to assign a POS Tag.

Hidden Markov Model:

- A finite set of states
  Here, for out problem, the states can be various parts of speech like noun, verb, adjectives, adverb, etc.

- A sequence of observations
  (terminals i.e the words in our sentence)
- Transition probability
  defined as the probability of a state 's' appearing after observing u and v in the sequence of observations
- Emission probability
  defined as the probability of making an observation x given that the state was s.

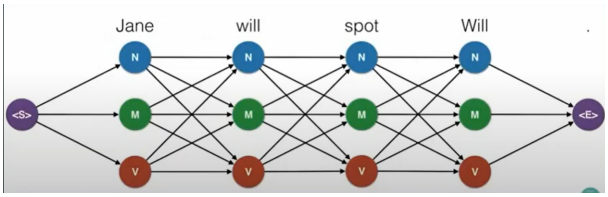

Fig. 9. POS tagging

Hidden States :

Let's considers the example of sunny day or rainy day. Given 7 days, a day can sunny or rainy and we are given the probabilities of mood of a person to be happy or sad. Here we consider Hidden Transitions. We are given probabilities of a sunny day to be followed by a rainy day and rainy to be followed by sunny and so on. These states are called transition state. And direct observation which we get based on persons' mood are called sequence of observations.
Transition Probability : The probability related to hidden state.
Emission probability : The probability related to visible states.

Input: a sentence $x_1 \dots x_n$, parameters $q(s|u,v)$ and $e(x|s)$.
Initialization: Set $\pi(0,*,*) = 1$, and $\pi(0,u,v) = 0$ for all (u,v) such that $u \neq *$ or $v \neq *$.
Algorithm:

- For $k = 1 \dots n$,
  - For $u \in \mathcal{K}, v \in \mathcal{K}$,

$$\pi(k,u,v) = \max_{w \in \mathcal{K}} (\pi(k-1,w,u) \times q(v|w,u) \times e(x_k|v))$$

- **Return** $\max_{u \in \mathcal{K}, v \in \mathcal{K}} (\pi(n,u,v) \times q(\text{STOP}|u,v))$

Fig. 10. Pseudocode for Viterbi

Viterbi Algorithm :

Take a sample sentence "Jane will spot Will." In NLP, whenever we consider a sentence for pos tagging, we append the start of sentence with "start" tag and end of the sentence with "end" tag. There are various paths which can be followed to obtain a tag sequence. So we can say that for very large sentence the brute force method will be hopelessly inefficient.

So, Insted of brute force approach, we will see that we can find the highest probable tag sequence efficiently using a dynamic programming algorithm known as the VITERBI algorithm. It basically stores the cost between any two nodes. So we don't have to calculate the cost every single point. Suppose I'm consider a path from NOUN followed MODEL AUXILIARY VERB and ADJECTIVE followed by NOUN. So I calculate it. Now again if I want to know the cost for NOUN followed by MODEL AUXILIARY VERB and MODEL AUXILIARY VERB followed by VERB, I don't have to recalculate the cost for NOUN followed by MODEL AUXILIARY VERB, I will already have the probability stored in the form of a dynamic programming table.

Fig 10 is the pseudocode of the Viterbi Algorithm. What it essentially does is, it multiplies the previous probabilities which we have obtained so far with the emission probability of the current sequence of observations. The maximum of previous and current one is taken to be the most efficient one.

Results of developed POS Tagger :

- Training the model :
  Using the POS tagged data from http://universaldependencies.org for training of POS tagger, the following results are obtained. Marathi POS tagging model :

```
start~tag~SYM:0.000141944641589779985805535841O
SYM~tag~NN:0.22095238095238095238095238l0
NN~tag~CC:0.025325884543761638733705772B1
CC~tag~NN:0.279863481228668941979522184З
NN~tag~NN:0.29590316573556797020484171З2
NN~tag~VM:0.3148975791433891992551210428
VM~tag~VAUX:0.311554078995939461055740125S
VAUX~tag~DEM:0.008973080757726819541375872383
DEM~tag~NN:0.827586206896551724137931034S
NN~tag~NNPC:0.041527001862197392923649906B9
NNPC~tag~NNP:0.711498973305954825462012320З
NNP~tag~PRP:0.210306406685236768802228412З
PRP~tag~NNP:0.058064516129032258064516129О3
NNP~tag~NN:0.217270194986072423398328690B
NN~tag~QO:0.004096834264432029795158286778
QO~tag~NN:0.546875
VM~tag~PUNC:0.087486157253599114064230340З0
PUNC~tag~SYM:0.070921985815602836879432624ll
SYM~tag~DEM:0.100952380952380952380952381O
NNPC~tag~NNPC:0.285420944558521560574948665З
PRP~tag~VM:0.143548387096774193548387096B
VM~tag~PUNC
:0.232188999630860059763750461
start~tag~JJ:0.000130259215839520646085710564O
JJ~tag~NNPC:0.006914433880726015557476231634
PRP~tag~QC:0.033064516129032258064516129О3
OC~tag~NN:0.69455252918287937743190661480
```

Fig. 11. Transmission probability

Fig. 11 and Fig. 12 represent the developed model itself. Fig. 11 shows the data stored after counting the transmission probability between two POS tags for e.g. NN tag CC:0.0253258845437616.... So this represents the probability of a word having POS tag NN followed by a word having tag CC is 0.0253258845437616... In Fig. 12 we get to see the Emission probability of a word having a particular POS tag for e.g अब-हइवअदअनअ_NN:0.000372439478584729981378026070B
So this represents that the probability of having 'NN' tag to the word अबहएएवअदअनअ is 0.000372439478584729981378026070B.

Fig. 13 represents the input for the testing. While testing, the **Viterbi Algorithm** is used to give the POS tags to testing data. This algorithm gives output in the form of all the words of the sentences are tagged with some POS tag which is shown in Fig. 14.

Fig. 16 shows the accuracy of the tagger after verifying it from Fig. 15 which contains actual expected output for input

```
वतीने_NN:0.00018621973929236499068901303 54
तसेच_PRP:0.00241935483870967741935483871 0
जर्नलिस्ट_NNPC:0.00102669404517453798767 9671458
फाउंडेशन_NNP:0.00069637883008356545961 00278552
ग्रामीण_NNPC:0.00102669404517453798767 9671458
सहभागाने_NN:0.00018621973929236499068 90130354
अभिवादन_NN:0.00037243947858472998137 80260708
कार्यक्रम_NN:0.00186219739292364990689 0130354
आयोजित_JJ:0.00518582541054451166810 7173725
केला_VM:0.01661129568106312292358803 987
होता_VAUX:0.02392821535393818544366 899302
परिषदेचे_NNP:0.00069637883008356545961 00278552
कार्याध्यक्ष_NN:0.00018621973929236499068 90130354
यशवंत_NNPC:0.00205338809034907597535 9342916
पाध्ये_NNP:0.00139275766016713091922 0055710
संघाचे_NNP:0.00278551532033426183844 0111421
अध्यक्ष_NN:0.00223463687150837988826 8156425
शशी_NNPC:0.00205338809034907597535 9342916
सावंत_NNP:0.00139275766016713091922 0055710
यावेळी_PRP:0.00161290322580645161290 3225806
उपस्थित_JJ:0.01469317199654278305963 699222
होते_VM:0.01513473606496862310815799188
श्री._NNPC:0.03490759753593429158110 882957
म्हणाले_VM:0.01107419712070874861572 535991
नव्या_JJ:0.01123595505617977528089887640
```

Fig. 12. Emission probability

```
तो माझा मित्र आहे .
मात्र , त्यांना गती नसल्याचे आढळून आले आहे .
त्यामुळे योजनांच्या कामांची केंद्रीय नगरविकास खात्याचे अधिकारी पाहणी करणार आहेत .
या पार्श्वभूमीवर योजनांमध्ये बदल करण्याचा निर्णय घेण्यात आला आहे .
त्यात , महापालिकेच्या योजनांची अंमलबजावणी त्यांच्या पातळीवर होणार आहे , तर ' पीएमपी 'ने आखलेल्या तीन प्रकल्पांच्या अंमलबजावणीची जबाबदारी ' पीएमपी ' कडेच असेल , असे सांगण्यात आले आहे .
प्रकल्पातील काही योजना वगळण्यात येणार असल्याने शिक्षण , आरोग्य , पर्यटन आणि पर्यावरणाचा योजनांमध्ये समावेश करण्यात येणार असल्याचे स्पष्ट करण्यात आले आहे .
जगताप म्हणाले , " या प्रकल्पातील काही योजनांचे स्वरूप बदलण्याचा प्रयत्न आहे .
त्यानुसार नव्या चार घटकांवर काम करण्यात येणार असून , त्याबाबत संचालक मंडळाच्या बैठकीत चर्चा करून निर्णय घेण्यात येतील .
त्या त्या पंत्रणांकडे योजना वर्ग केल्यानंतर त्यांच्या खर्चात कपात होणार आहे .
तो खर्च नव्या योजनांकरिता वापरण्यात येईल .
विशाष विश्वास , आरोग्य आणि पर्यावरण या क्षेत्रातील कामांसाठी नव्याने काही तरतूदही करण्याचे नियोजन आहे .
योजना निश्चित झाल्यानंतर त्यांच्या अंमलबजावणीच्या दृष्टीने कार्यवाही करू .
" स्मार्ट सिटी प्रकल्पातील योजनांचा आढावा घेण्यासाठी येत्या दहा नोव्हेंबरला संचालक मंडळाची बैठक होणार आहे .
बैठकीत , सल्लागार कंपनी असलेल्या ' मॅकेन्झी ' बाबत निर्णय घेतला जाण्याची शक्यता आहे .
```

Fig. 13. Marathi Test Input

```
Output :
तो_PRP माझा_PRP मित्र_NN आहे_VM .
_VM
मात्र_RP ,_PUNC त्यांना_PRP गती_NN नसल्याचे_VM आढळून_VM आले_VAUX आहे_VAUX .
_VAUX
त्यामुळे_CC योजनांच्या_NNPC कामांची_NNPC केंद्रीय_NNPC नगरविकास_NNP खात्याचे_PRP अधिकारी_NN पाहणी_NN करणार_VM आहेत_VAUX .
_VAUX
या_DEM पार्श्वभूमीवर_NN योजनांमध्ये_NN बदल_NN करण्याचा_VM निर्णय_NN घेण्यात_VM आला_VAUX आहे_VAUX .
_VAUX
त्यात_PRP ,_PUNC महापालिकेच्या_NNP योजनांची_PRP अंमलबजावणी_NN त्यांच्या_PRP पातळीवर_NN होणार_VM आहे_VAUX ,_PUNC तर_CC '_NNPC पीएमपी_NNP 'ने_PRP आखलेल्या_NN तीन_QC प्रकल्प
_VAUX
प्रकल्पातील_UT काही_QF योजना_NN वगळण्यात_VM येणार_VAUX असल्याने_VAUX शिक्षण_NN ,_PUNC आरोग्य_NNP ,_PUNC पर्यटन_NNPC आणि_CC पर्यावरणाचा_NNC योजनांमध्ये_NN समावेश_NN करण्यात
_VAUX
जगताप_NNP म्हणाले_VM ,_PUNC "_SYM या_DEM प्रकल्पातील_NN काही_QF योजनांचे_NN स्वरूप_NN बदलण्याचा_NN प्रयत्न_NN आहे_VM .
_VM
त्यानुसार_PRP नव्या_JJ चार_QC घटकांवर_NN काम_NN करण्यात_VM येणार_VAUX असून_VAUX ,_PUNC त्याबाबत_PRP संचालक_NN मंडळाच्या_NN बैठकीत_NN चर्चा_NN करून_VM निर्णय_NN घेण्यात_VM
_VAUX
त्या_PRP त्या_DEM पंत्रणांकडे_NN योजना_NN वर्ग_NN केल्यानंतर_VM त्यांच्या_PRP खर्चात_NN कपात_NN होणार_VM आहे_VAUX .
_VAUX
तो_PRP खर्च_NN नव्या_JJ योजनांकरिता_NN वापरण्यात_VM येईल_VAUX .
_VAUX
विशाष_CC विश्वास_NN ,_PUNC आरोग्य_SYM आणि_CC पर्यावरण_PUNC या_DEM क्षेत्रातील_NN कामांसाठी_NN नव्याने_JJ काही_QF तरतूदही_NN करण्याचे_VM नियोजन_NN आहे_VM .
_VM
योजना_NN निश्चित_JJ झाल्यानंतर_VM त्यांच्या_PRP अंमलबजावणीच्या_NN दृष्टीने_NN कार्यवाही_NN करू_VM .
_VM
" _SYM स्मार्ट_NNPC सिटी_NNP प्रकल्पातील_PRP योजनांचा_NN आढावा_NN घेण्यासाठी_VAUX येत्या_VM दहा_QC नोव्हेंबरला_NN संचालक_NN मंडळाची_NN बैठक_NN होणार_VM आहे_VAUX .
_VAUX
बैठकीत_NN ,_PUNC सल्लागार_DEM कंपनी_NN असलेल्या_VM '_VAUX मॅकेन्झी_PUNC '_DEM बाबत_NN निर्णय_NN घेतला_VM जाण्याची_VAUX शक्यता_NN आहे_VM ._PUNC
```

Fig. 14. Marathi Test Output

```
मात्र_CC ,_PUNC त्यांना_PRP गती_NN नसल्याचे_VM आढळून_VM आले_VM आहे_VAUX ._PUNC
त्यामुळे_CC योजनांच्या_NN कामांची_NN केंद्रीय_NN नगरविकास_NN खात्याचे_NN अधिकारी_NN पाहणी_VM करणार_VM आहेत_VAUX ._PUNC
या_DM पार्श्वभूमीवर_NN योजनांमध्ये_NN बदल_NN करण्याचा_VM निर्णय_NN घेण्यात_VM आला_VM आहे_VAUX ._PUNC
त्यात_RB ,_PUNC महापालिकेच्या_NN योजनांची_NN अंमलबजावणी_NN त्यांच्या_PRP पातळीवर_NN होणार_VM आहे_VAUX ,_PUNC तर_CC '_SYM पीएमपी_UNK '_SYM ने_DM आखलेल्या_VM तीन_QTC प्रकल्पांच्या_N
प्रकल्पातील_NN काही_QTF योजना_NN वगळण्यात_VM येणार_VM असल्याने_VAUX शिक्षण_NN ,_PUNC आरोग्य_NN ,_PUNC पर्यटन_NN आणि_CC पर्यावरणाचा_NN योजनांमध्ये_NN समावेश_NN करण्यात_VM येणार_VM
जगताप_NNP म्हणाले_VM ,_PUNC "_SYM या_DM प्रकल्पातील_NN काही_QTF योजनांचे_NN स्वरूप_NN बदलण्यात_VM प्रयत्न_NN आहे_VAUX ._PUNC
त्यानुसार_RB नव्या_JJ चार_QTC घटकांवर_NN काम_NN करण्यात_VM येणार_VM असून_VAUX ,_PUNC त्याबाबत_NN संचालक_NN मंडळाच्या_NN बैठकीत_NN चर्चा_NN करून_VM निर्णय_NN घेण्यात_VM येतील_VAUX
त्या_DM त्या_DM पंत्रणांकडे_NN योजना_NN वर्ग_NN केल्यानंतर_VM त्यांच्या_PRP खर्चात_NN कपात_NN होणार_VM आहे_VAUX ._PUNC
तो_PRP खर्च_NN नव्या_JJ योजनांकरिता_NN वापरण्यात_VM येईल_VAUX ._PUNC
विशाष_RB विश्वास_NN ,_PUNC आरोग्य_NN आणि_CC पर्यावरण_NN या_DM क्षेत्रातील_NN कामांसाठी_NN नव्याने_JJ काही_QTF तरतूदही_NN करण्याचे_VM नियोजन_NN आहे_VAUX ._PUNC
योजना_NN निश्चित_JJ झाल्यानंतर_VM त्यांच्या_PRP अंमलबजावणीच्या_NN दृष्टीने_NN कार्यवाही_NN करू_VM ._PUNC
" _SYM स्मार्ट_UNK सिटी_UNK प्रकल्पातील_NN योजनांचा_NN आढावा_NN घेण्यासाठी_VM येत्या_VM दहा_QTC नोव्हेंबरला_NNP संचालक_NN मंडळाची_NN बैठक_NN होणार_VM आहे_VAUX ._PUNC
बैठकीत_NN ,_PUNC सल्लागार_JJ कंपनी_NN असलेल्या_VM '_SYM मॅकेन्झी_UNK '_SYM बाबत_NN निर्णय_NN घेतला_VM जाण्याची_VM शक्यता_NN आहे_VAUX ._PUNC
```

Fig. 15. Marathi Expected Output

```
Wrong Predictions =  84
Total Predictions =  208
Accuracy is =  59.61538461538461 %
```

Fig. 16. Accuracy of Marathi POS tagger model on test data

data shown in Fig. 13.

Similarly, the model can generate Transmission probability and emission probability based model for any language given the POS tagged train-data and can give output based on the generated model.

D. NMT using simple seq2seq model

In our model, we simplely create neural machine translator to translate Marathi sentences into hindi sentences, by applying the Long Short Term Memory (LSTM) model. In deep learning, we make use of neural networks to train our deep learning model(that makes predictions) so there are some neural networks like RNN, LSTM, GRU that works on a sequence as inputs like a sequence of English sentence in our case. So Seq2Seq model is a representation of these special neural networks. The Seq2Seq model has two parts one is called an encoder and the other is called a decoder and as the name suggests encoder encodes our input sentences and decoder decodes these encoded sentences into output sentences. So this is a simple introduction of a simple seq2seq model and based on further improvements and research we keep on finding new techniques to improve the encoding of the input sequence but the heart of seq2seq model is encoder and decoder. The seq2seq model wants all the input sequence to be of same length so one way to achieve this is, we calculate the length of each sentence in both corpus and then we choose the max length i.e the longest sentence in both Hindi and Marathi thus, in the end, we have two max_length one for Hindi sentence and another is for Marathi sentence. After this, since machines understand numbers only and not the text we need a way to convert the input text sequence into number and one such way is to index the words of the sentences. We can do this indexing using the tokenizer. We also need to take the vocabulary size of both Hindi and Marathi corpus which is required when we create the input data for the model training. We train our encoder and decoder models with the required data using the 'Teacher Forcing' technique and fit the model by taking appropriate hyper-parameters. The best weights are saved and the trained model is later saved in JSON format for prediction and accuracy testing purposes.

We also try to predict outputs from our model and compare it with google translate prediction.

IV. RESULTS AND CONCLUSION

We have achieved a good accuracy of around 59.61% on our POS tagging approach, and our model performs at an accuracy of 66.83% on the training data. We have also successfully implemented an LSTM model for our Indic data

```
Epoch 41/50
271/271 [==============================] - 18s 66ms/step - loss: 0.5549 - accuracy: 0.6334
Epoch 42/50
271/271 [==============================] - 18s 67ms/step - loss: 0.5432 - accuracy: 0.6379
Epoch 43/50
271/271 [==============================] - 18s 67ms/step - loss: 0.5323 - accuracy: 0.6419
Epoch 44/50
271/271 [==============================] - 18s 67ms/step - loss: 0.5216 - accuracy: 0.6465
Epoch 45/50
271/271 [==============================] - 18s 67ms/step - loss: 0.5114 - accuracy: 0.6501
Epoch 46/50
271/271 [==============================] - 18s 67ms/step - loss: 0.5013 - accuracy: 0.6541
Epoch 47/50
271/271 [==============================] - 18s 67ms/step - loss: 0.4919 - accuracy: 0.6579
Epoch 48/50
271/271 [==============================] - 18s 67ms/step - loss: 0.4824 - accuracy: 0.6616
Epoch 49/50
271/271 [==============================] - 18s 67ms/step - loss: 0.4734 - accuracy: 0.6651
Epoch 50/50
271/271 [==============================] - 18s 67ms/step - loss: 0.4649 - accuracy: 0.6683
```

Fig. 17. Accuracy of model on training data

```
Test sentance:  2
sentance:  मी त्याच्याशी एकट्याने बोलेन
origianl translate: मैं अकेले उससे बात करूंगा।
predicted Translate:  मैं पिछले हफ्ते अकेला नहीं था।
==================================================
Test sentance:  3
sentance:  तू आत्ता एकटी आहेस का
origianl translate: क्या आप अभी अकेले हैं
predicted Translate:  क्या आप अब अकेले हैं
==================================================
Test sentance:  4
sentance:  त्यांचे लांब केस आहेत
origianl translate: उसके लंबे बाल है।
predicted Translate:  वह लंबे बाल है।
==================================================
Test sentance:  5
sentance:  त्या कोण
origianl translate: वह कौन है
predicted Translate:  वो कौन है
==================================================
Test sentance:  6
sentance:  माझा दादा शिक्षक आहे
origianl translate: मेरा बड़ा भाई एक शिक्षक है।
predicted Translate:  मेरा भाई एक शिक्षक है।
==================================================
Test sentance:  7
sentance:  ती आपली शाळा आहे
origianl translate: वह हमारा स्कूल है।
predicted Translate:  वह मेरा प्रेमी है।
```

Fig. 18. Accuracy of model on training data

NMT model.Some limitations that we faced were due to long sentences and less and sparse training data that led to some hindrances in recording even better accuracy.

REFERENCES

[1] Laskar, S. R., Khilji, A. F. U. R., Pakray, P., Bandyopadhyay, S. (2020, November). Hindi-marathi cross lingual model. In Proceedings of the Fifth Conference on Machine Translation (pp. 396-401).

[2] Ojha, A. K., Rani, P., Bansal, A., Chakravarthi, B. R., Kumar, R., McCrae, J. P. (2020, November). NUIG-Panlingua-KMI Hindi-Marathi MT Systems for Similar Language Translation Task@ WMT 2020. In Proceedings of the Fifth Conference on Machine Translation (pp. 418-423).

[3] Mujadia, V., Sharma, D. M. (2020, November). Nmt based similar language translation for hindi-marathi. In Proceedings of the Fifth Conference on Machine Translation (pp. 414-417).

[4] Pal, S., Zampieri, M. (2020, November). Neural machine translation for similar languages: The case of indo-aryan languages. In Proceedings of the Fifth Conference on Machine Translation (pp. 424-429).

[5] Kumar, A., Baruah, R., Mundotiya, R. K., Singh, A. K. (2020, November). Transformer-based Neural Machine Translation System for Hindi–Marathi: WMT20 Shared Task. In Proceedings of the Fifth Conference on Machine Translation (pp. 393-395).

[6] Kumar, A., Baruah, R., Mundotiya, R. K., Singh, A. K. (2020, November). Transformer-based Neural Machine Translation System for Hindi–Marathi: WMT20 Shared Task. In Proceedings of the Fifth Conference on Machine Translation (pp. 393-395).

[7] Laskar, S. R., Pakray, P., Bandyopadhyay, S. (2019, August). Neural machine translation: Hindi-Nepali. In Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2) (pp. 202-207).

[8] Przystupa, M., Abdul-Mageed, M. (2019, August). Neural machine translation of low-resource and similar languages with backtranslation. In Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2) (pp. 224-235).

[9] Manger, K. M. (2014). Translation Divergences in Hindi-Nepali Machine Translation. Language in India, 14(5).

[10] Liu, Y., Sun, C., Lin, L., Wang, X. (2016). Learning natural language inference using bidirectional LSTM model and inner-attention. arXiv preprint arXiv:1605.09090.