# ADVANCED NLP - PROJECT REPORT

## TRANSLITERATION

Team Multilinguals_1394 (28) : Sriharshitha B (2018111013), Samartha SM (2018101094)
Mentor: Mani Kanta

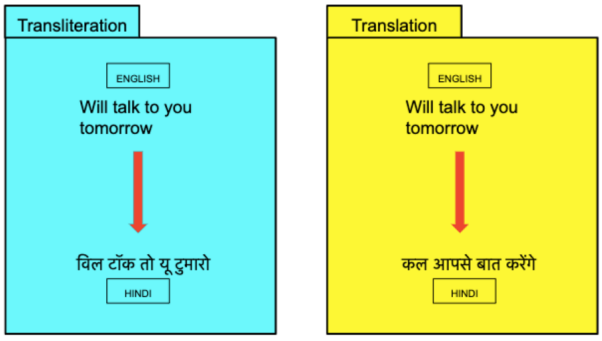
---

## What is Transliteration?

The mechanism of converting a word from a source language to a target language such that the output word

- is phonemically equivalent to the source language word,
- conforms to the phonology of the target language
- matches the user intuition of the equivalent of the source language word in the target language

In machine translation, the objective is to preserve the semantic meaning of the utterance as much as possible while following the syntactic structure in the target language. In Transliteration, the objective is to preserve the original pronunciation of the source word as much as possible while following the phonological structures of the target language. Simply put, writing a word written in one language using the alphabet of the second language.



Domain: Transliteration Description: Transliterate Hindi language from Devanagari script to Roman Script (English).

## Dataset?

Dataset: Dakshina (https://github.com/google-research-datasets/dakshina)
The Dakshina dataset is a collection of text in both Roman and native scripts for 12 South Asian languages. For each language, the dataset includes a large collection of native script Wikipedia text, a romanization lexicon which consists of words in the native script with attested romanizations, and some full sentence parallel data in both a native script of the language and the basic Latin alphabet.

```
इसके आने से पूर्व ही लोग घरों की सफाई का कार्य शुरू कर देते हैं।
विलुप्ति की कगार पर गुणकारी तीखुर
```

```
iske aane se purva hi log gharon ki safai ka karya shuru kar dete hain.
vilupti ki kagaar par gunkaari tikhur
```

**Text cleaning/Pre-processing:**

As mentioned above, the dataset consisted of parallel sentences in english and hindi in 2 files.

| | roman | native |
|---|---|---|
| 0 | इसके आने से पूर्व ही लोग घरों की सफाई का कार्य... | iske aane se purva hi log gharon ki safai ka k... |
| 1 | विलुप्ति की कगार पर गुणकारी तीखुर\n | vilupti ki kagaar par gunkaari tikhur\n |
| 2 | माइकल कामेन (द वॉल के वाद्यमय हिस्सों के लिए ए... | Michael kamen (the wall ke vadyamaya hisso ke ... |
| 3 | शाहनामा नौरोज़ के त्यौहार को महान जमशेद के शास... | shahnama noroj ke tyohaar ko mahaan Jamshed ke... |
| 4 | मेहरोत्रा, डॉ॰ एन.\n | Mehrotra, Dr॰ N.\n |

As a part of data preprocessing, after reading thos files, we have and preprocessed both the data in the same way as we want word to word matching. Following is the code snippet for the same.

```python
def preprocess_sentence_native(w):
    w = w.lower().strip()
    ww = ''
    # Removing punctuations
    ww += ''.join(ch for ch in w if ch not in string.punctuation)
    # Remove all numbers from text
    remove_digits = str.maketrans('', '', digits)
    ww = ww.translate(remove_digits)
    ww = ww.rstrip().strip()
    # Adding start and end tags
    ww = '@' + ww + '#'
    return ww

def preprocess_sentence_roman(w):
    w = w.lower()
    cleaned_line = ''
    # Removing punctuations
    cleaned_line += ''.join(ch for ch in w if ch not in string.punctuation)
    # Remove all numbers from text
    remove_digits = str.maketrans('', '', digits)
    cleaned_line = cleaned_line.translate(remove_digits)
    # Removing numbers written in roman script and other characters
    cleaned_line = re.sub("[२३०८९५७१४६।–]", "", cleaned_line)
    cleaned_line = cleaned_line.rstrip().strip()
    # Adding start and end tags
    cleaned_line = '@' + cleaned_line + '#'
    return cleaned_line
```

We took each preprocessed sentence both in roman and native scripts and split it to get the words in the sentence. Words in both scripts have to be mapped to each other. Thus, we eliminated the sentences like following. ( Parallel sentences that do not have equal number of words.)

```
@दक्षिण अफ्रीकी महिला क्रिरकेट टीम का न्यूज़ीलैंड दौरा#  -  @dakshin afriki mahila kriket teem ka new zealand dauraa#
```

Here, when we usually construct vocabulary for NLP tasks, we eliminate the repetitions. But, we haven't done that here. Because, one word in roman script might be transliterated in multiple ways (noise in data). So, to capture that minute details, and to not lose the correct information, we have not eliminated the repetitions. Following are the words obtained from the preprocessed sentences.

```
['@इसके', 'आने', 'से', 'पूर्व', 'ही', 'लोग', 'घरों', 'की', 'सफाई', 'का'],
['@iske', 'aane', 'se', 'purva', 'hi', 'log', 'gharon', 'ki', 'safai', 'ka'])
```

Above preprocessing is done for both test and train data. The whole data is then combined to then make a reasonable train-test split.

**Creating input and target tensors**

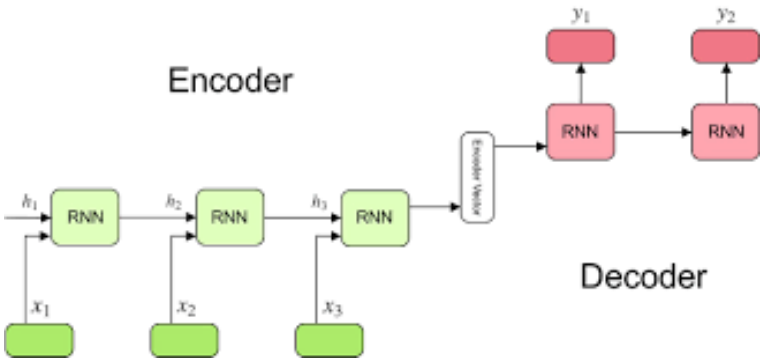
```
convert(inp_lang,input_tensor[-1])
convert(targ_lang,target_tensor[-1])
```

```
70 ----> म
73 ----> ल
96 ----> ◌ि
80 ----> ह
94 ----> ◌ो
60 ----> त
96 ----> ◌ि
72 ----> र
82 ----> ◌ा
1 ----> #
15 ----> m
3 ----> a
14 ----> l
10 ----> h
17 ----> o
22 ----> t
20 ----> r
3 ----> a
1 ----> #
```

**Baseline Model**

The baseline model which we implemented in this project will be a  sequence - to - sequence (seq2seq) model with encoder - decoder architecture to model the mappings from encoded data to decoded data. The encoder accumulates the hidden meaning and position information of the input tensors and the decoder uses this hidden information with the primer to predict the next character in the target language. As transliteration deals with the character positioning in the transliterated words, the input data will be batches of character embeddings.

**Train Words** : 128765
**Validation Words** : 42922

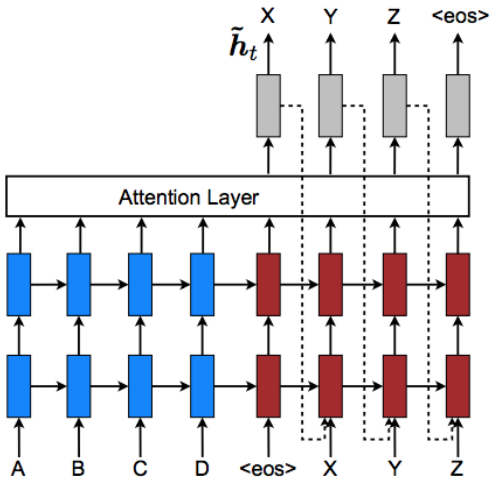



Encoder output shape: (batch size, sequence length, units) (128, 32, 1024)
Encoder Hidden state shape: (batch size, units) (128, 1024)
Decoder output shape: (batch_size, vocab size) (128, 45)

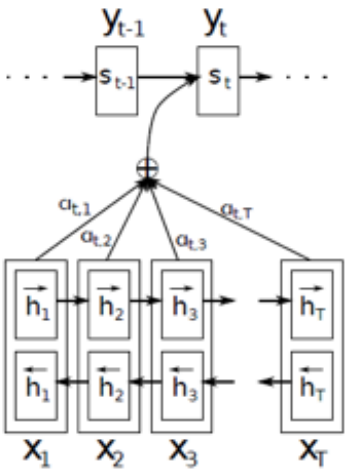
## Improvisation (Baseline+ Model)

The baseline model can be improved by using attention (BahdanauAttention) mechanism in the encoder - decoder network, which will improve the transliteration task by learning better representations. Attention captures the importance of specific characters in the input language for each character in the target language using the context representations which highlights the relevant information/features from the input data through hidden representations.

**Train Words** : 90000
**Validation Words** : 10000



Attention result shape: (batch size, units) (128, 1024)
Attention weights shape: (batch_size, sequence_length, 1) (128, 32, 1)




## Training

Optimiser: Adam
Loss function: SparseCategoricalCrossentropy
Epochs: Baseline trained for 20 epochs and Baseline+ trained for 10 epochs
Batch_size: 128
Embedding_dim: 256
Units: 1024

Context and target sequences prepared are fed to the model and trained. Loss per epoch is calculated and the following is the graph of the same. Loss per epoch is decreasing and this clearly indicates that the model is learning and it is converging.

## Evaluation

Metrics: Accuracy, BLEU score and Rouge score

***BLEU score:***

The Bilingual Evaluation Understudy Score, or BLEU for short, is a metric for evaluating a generated sentence to a reference sentence. It is used to evaluate MT models. It is also used in Language generation, Image caption generation, Text summarization, Speech recognition etc. 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. It is not perfect but correlates highly with human evaluation and has been widely adopted.

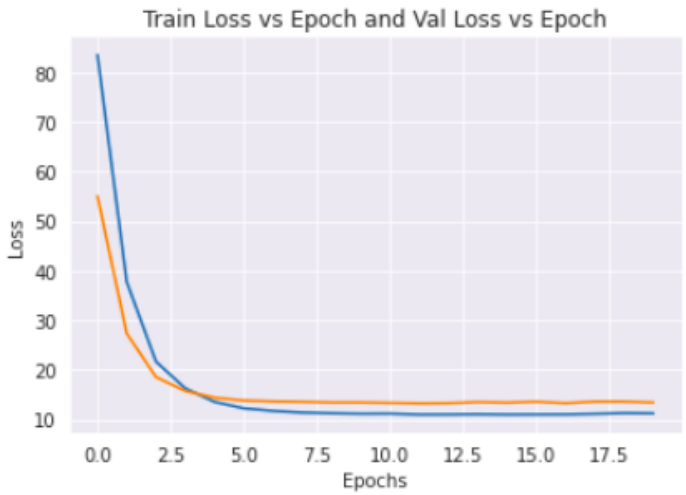
The approach works by counting matching n-grams in the candidate translation to n-grams in the reference text, where 1-gram or unigram would be each token and a bigram comparison would be each word pair. The comparison is made regardless of word order. We have computed the bleu score for the generated transliterated sentences and the ground truth.

***Rouge Score:***

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scoring algorithm calculates the similarity between a candidate document and a collection of reference documents. The ROUGE score is used to evaluate the quality of document translation and summarization models. It is a set of metrics which is important in evaluating the overall model. Some of the metrics which is present in ROUGE are ROUGE-N, ROUGE-L, Precision, Recall, F1 Score, ROUGE-S. ROUGE-N measures the number of matching ngrams, whereas ROUGE-L measures Longest Common Subsequence (LCS). Precision, Recall and F1 score are important metrics in ML, which evaluates the performance of the model. However, the drawbacks of Rouge score is that it does not consider different words with meaning, as it only looks at the complete string matching in the documents.

**Outputs**

1) **Model 1 - Baseline (Input: Native, Output: Roman)**
   Loss:




Accuracy:

```
100% 10/10 [00:00<00:00, 13.57it/s]
Accuracy on 10 test words: 0.7

100% 100/100 [00:06<00:00, 17.98it/s]
Accuracy on 100 test words: 0.51

100% 500/500 [00:32<00:00, 19.16it/s]
Accuracy on 500 test words: 0.434

100% 1000/1000 [01:05<00:00, 14.87it/s]
Accuracy on 1000 test words: 0.415

100% 5000/5000 [05:21<00:00, 17.68it/s]
Accuracy on 5000 test words: 0.4224
```

Bleu score and Rouge score:

```
Bleu score on 1000 test sentences: 0.22525991214632998

Rouge score on 1000 test sentences: (0.4387891778934867, 0.4387891778934867, 0.4387891778934867)
```

*Rouge score is of the format (precision, recall, f1 score)

Following are some of the transliterations obtained for different words.

```
Input:  वाम
Output:  vam#
```

```
Input:  ऐडसेंस
Output:  edcems#
```

```
Input:  कायदा
Output:  yadayaa
```

```
Input:  गुणदोषों
Output:  guandarashon#
```

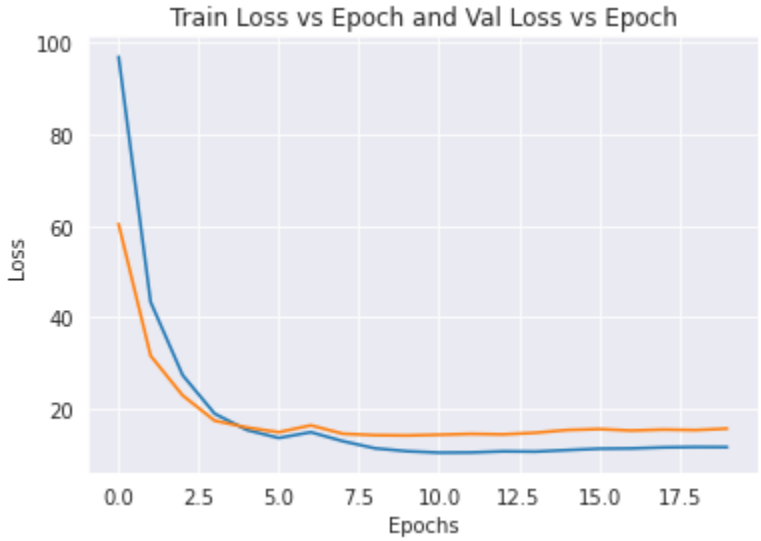
```
Input:  संस्थागत
Output:  sansthaar#
```

Following are some of the transliterations obtained for different sentences.

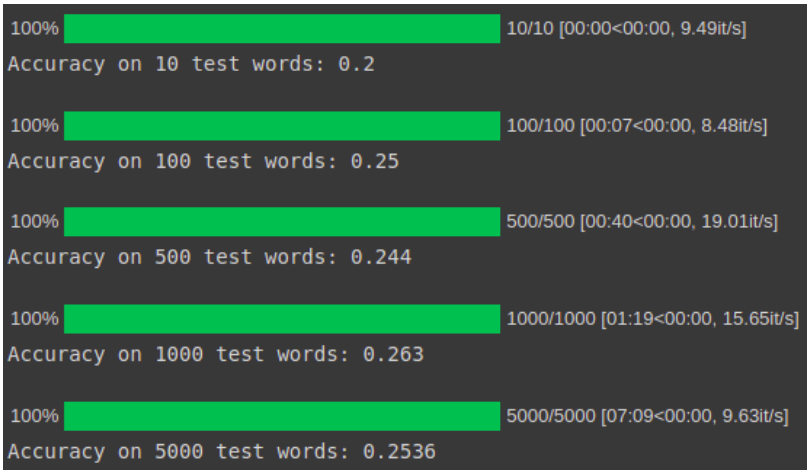
```
Input:  इसके आने से पूर्व ही लोग घरों की सफाई का कार्य शुरू कर देते हैं।
Output:  iske# aane# se# purva# hi# loga# ghaharan# ki# safai# ka# carya# shushra# kar# detai# hain#

Input:  शाहनामा नौरोज़ के त्यौहार को महान जमशेद के शासनकाल से जोड़ता है।
Output:  shahanatama# nauroz ken# tyuhara# ko# mahan# jamesh# ken# sashasanahal se# jomata# ha#

Input:  सारे लहलहाते खेत, पेड़-पौधे, महल और झोपड़ियाँ- सब की सब उजड़ जाएँगी।
Output:  saare# lahalate# khet# pedeppay# mahal# aur# gopameena# saba# ki# saba# udaza# jagayn#

Input:  माना जाता है की मरीज की अपनी ही प्रतिरक्षा प्रणाली के क्षति की जाती है।
Output:  mana# jata# ha# ki# merice# ki# apni# hi# prashtriksha pranali# ken# kshit# ki# jati# ha#

Input:  इस युद्ध में राणा सांगा का साथ महमूद लोदी दे रहे थे।
Output:  is# yuddh# men# rana# sanga# ka# saath# mahmud# loydi# de# rahe# the#
```

2) **<u>Model 2</u> - Baseline (Input: Roman, Output: Native)**

Loss:



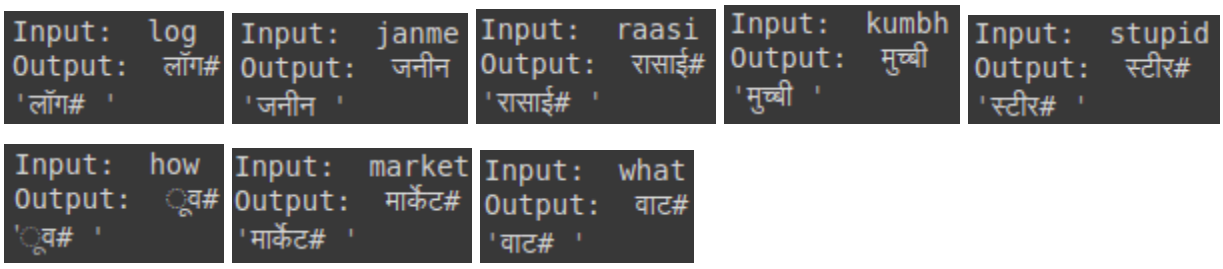

Accuracy:



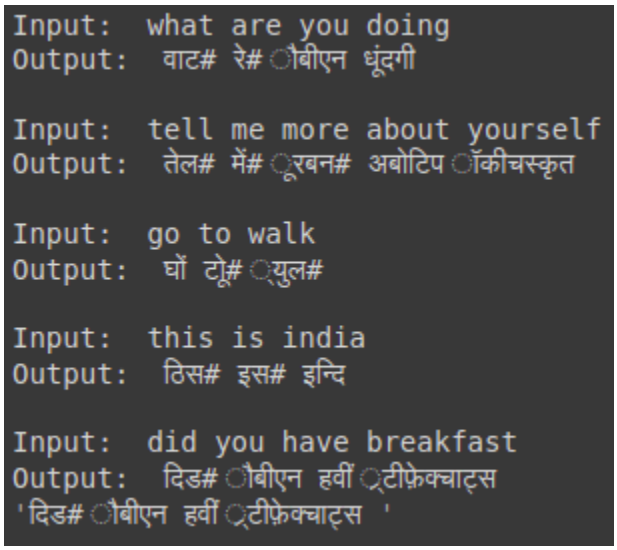

Bleu score:

```
Bleu score on 1000 test sentences: 0.18506658724406877
```

Following are some of the transliterations obtained for different words:



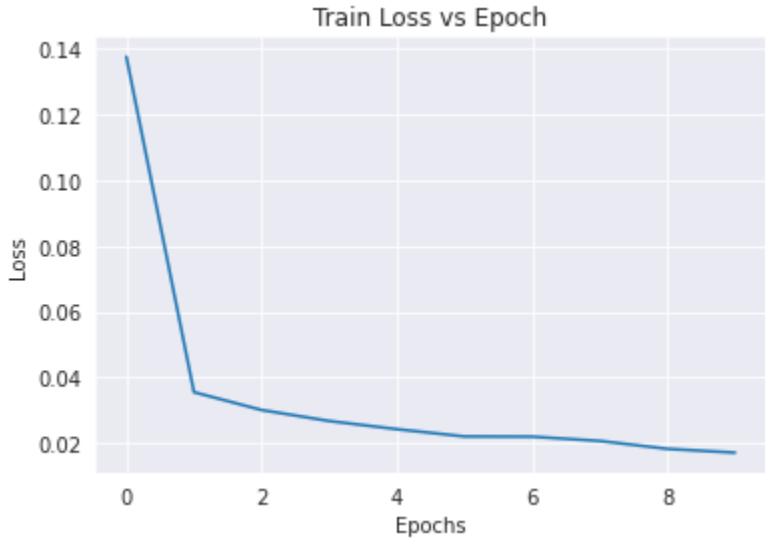

Following are some of the transliterations obtained for different sentences:



```
Input:  bogra ka yudh bharat paak yudh १९७१ ka bhag tha jo vartaman bangladesh mein hua tha
Output:  भोग्रा# का# ◌्यौढ़ भरा# पका# ◌्यौढ़ ◌े का# भग# था# ◌ूज़ो# वैरवाता बंग्लीशियन में# ◌ूहा# था#

Input:  taruna nirankari  richa sharma jaya ki dusri badi bahan
Output:  तुराना# निर्दन# रिचार शर्मा# जय# कीं# दूसरी# बीं# बाहान#

Input:  teen varsh se bacchhon ki bhasha ka vikaas hota hai
Output:  टीन# वर्ष# शे बच्चों# कीं# भाष# का# ◌्कींस# ◌ूठिया हि#

Input:  lili patel  savitri devi jaya ki nani
Output:  लिल# पेटेल# सरित्वी# देवी# जय# कीं# ननी#

Input:  ismen bbq sauce jodkar germany mein sthayi rup se beja chata hai
Output:  इसमें ब्लै# सुचेर झोड़कर# ◌लियंन्कर में# थली# रुप# शे बेज# क्ता# हि#
'इसमें ब्लै# सुचेर झोड़कर# ◌लियंन्कर में# थली# रुप# शे बेज# क्\u200dता# हि# '
```

### 3) <u>Model 3</u> - Baseline+ (Input: Native, Output: Roman)

Loss:




Accuracy:

```
100% 10/10 [00:01<00:00, 10.04it/s]
Accuracy on 10 test words: 0.9

100% 100/100 [00:10<00:00, 10.69it/s]
Accuracy on 100 test words: 0.83

100% 500/500 [00:52<00:00, 10.38it/s]
Accuracy on 500 test words: 0.766

100% 1000/1000 [01:46<00:00, 8.68it/s]
Accuracy on 1000 test words: 0.782

100% 5000/5000 [08:54<00:00, 8.48it/s]
Accuracy on 5000 test words: 0.7516
```

Bleu scores and Rouge score:

```
Bleu score on 1000 test words: 0.4845468108088249

Rouge score on 1000 test words: (0.7635816144639673, 0.7635816144639673, 0.7635816144639673)
```

*Rouge score is of the format (precision, recall, f1 score)

Following are some of the best transliterations obtained for different words.

```
Input:  संस्थागत
Output:  sansthagat#
```

```
Input:  गुणदोषों
Output:  gundoshon#
```

```
Input:  कायदा
Output:  kayada#
```

```
Input:  ऐडसेंस
Output:  aidsens#
```

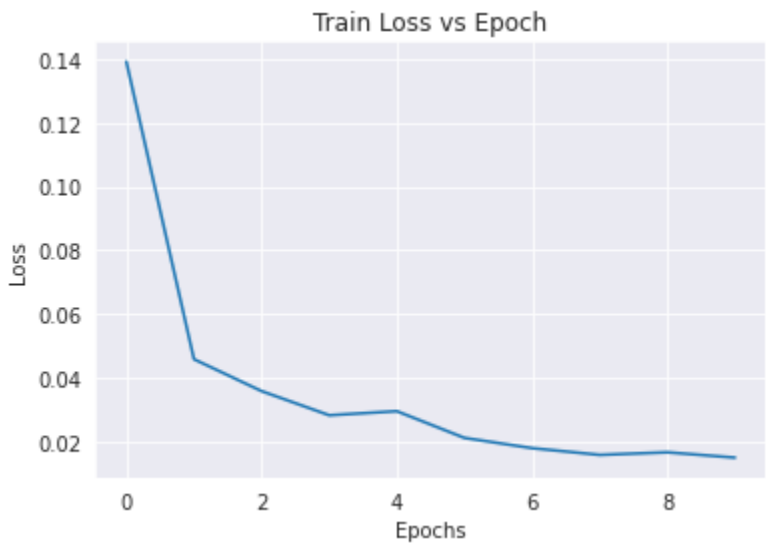
```
Input:  वाम
Output:  vam#
```

Following are some of the transliterations obtained for different sentences.

```
Input:  इसके आने से पूर्व ही लोग घरों की सफाई का कार्य शुरू कर देते हैं।
Output:  iske# ane# se# purv# hi# log# gharon# ki# safai# ka# karya# shuru# kar# dete# hain#

Input:  शाहनामा नौरोज़ के त्यौहार को महान जमशेद के शासनकाल से जोड़ता है
Output:  shahanama# nauroz# ke# tyauhar# ko# mahan# jamshed# ke# shasankaal# se# jodta# hai#

Input:  सारे लहलहाते खेत, पेड़-पौधे, महल और झोपड़ियाँ- सब की सब उजड़ जाएँगी।
Output:  sare# lahalahate# khet# pedpaudhe# mahal# aur# jhopdiyan# sab# ki# sab# ujd# jayengi#

Input:  माना जाता है की मरीज की अपनी ही प्रतिरक्षा प्रणाली के क्षति की जाती है।
Output:  mana# jata# hai# ki# marij# ki# apni# hi# pratiraksha# pranali# ke# kshati# ki# jati# hai#

Input:  इस युद्ध में राणा सांगा का साथ महमूद लोदी दे रहे थे।
Output:  is# yuddh# men# rana# sanga# ka# sath# mahmud# lodi# de# rahe# the#
```

4) **<u>Model 4</u> - Baseline+ (Input: Roman, Output: Native)**

Loss:



Accuracy:

```
Accuracy on 10 test words: 0.8

Accuracy on 100 test words: 0.88

Accuracy on 500 test words: 0.788

Accuracy on 1000 test words: 0.774

Accuracy on 5000 test words: 0.7384
```

Bleu score:

```
Bleu score on 1000 test words: 0.6337390510214915
```

Following are some of the transliterations obtained for different words.

```
Input:  how
Output: हॉव#

Input:  market
Output: मरकेट#

Input:  stupid
Output: स्टूपाइड़#

Input:  kumbh
Output: कुंभ#

Input:  raasi
Output: रासी#

Input:  janme
Output: जन्मे#

Input:  log
Output: लोग#

Input:  what
Output: भाट#
```

Following are some of the transliterations obtained for different sentences.

```
Input:  what are you doing
Output:  भाट# आरे# यो# दोइंग#

Input:  tell me more about yourself
Output:  ेटल# में# मोरे# अबोट# यूर्सेल्फ़फ़फ़#

Input:  go to walk
Output:  गो# तो# वाल्क#

Input:  this is india
Output:  दीस# इस# इंदिया#

Input:  did you have breakfast
Output:  दीड# यो# हवे# ब्रेक्सास्ट#
```

```
Input:  bogra ka yudh bharat paak yudh १९७१ ka bhag tha jo vartaman bangladesh mein hua tha
Output:  बोग्रा# का# युद# भारत# पाक# युद# # का# भग# था# जो# वर्तमान# बैंग्लादेश# में# हुआ# था#

Input:  taruna nirankari  richa sharma jaya ki dusri badi bahan
Output:  तरुणा# निरंकरी# रिचा# शर्मा# जया# की# दूसरी# बाड़ी# बहन#

Input:  teen varsh se bacchhon ki bhasha ka vikaas hota hai
Output:  तीन# वर्ष# से# बच्चों# की# भाषा# का# विकास# होता# है#

Input:  lili patel  savitri devi jaya ki nani
Output:  लिली# पटेल# सवित्री# देवी# जया# की# नानी#

Input:  ismen bbq sauce jodkar germany mein sthayi rup se beja chata hai
Output:  इसमें# bbcbbbbbbbbbbbbbbbbbbbbbbbbbbb सौक# जोड़कर# गेर्मीय# में# स्थायी# रूप# से# बेजा# चाता# है#
```

**Drive Link:**

**https://drive.google.com/drive/folders/18A_XkgrtaX0XqdvwAcpCPzJ26RiF1AQp?usp=sharing**

Above is the link for model checkpoints

**References:**


1. https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346
2. https://towardsdatascience.com/sequence-2-sequence-model-with-attention-mechanism-9e9ca2a613a#:~:text=Seq2Seq%20model%20with%20an%20attention,Context%20vector
3. https://bsantraigi.github.io/tutorial/2019/08/31/english-to-hindi-transliteration-using-seq2seq-model.html