Two methods of final work :

1st:

Text

Description automatically generated

Reading dictionary and creating vocab and saving it into pkl file so that it wont be needing to create vocab everytime we run the program

Text

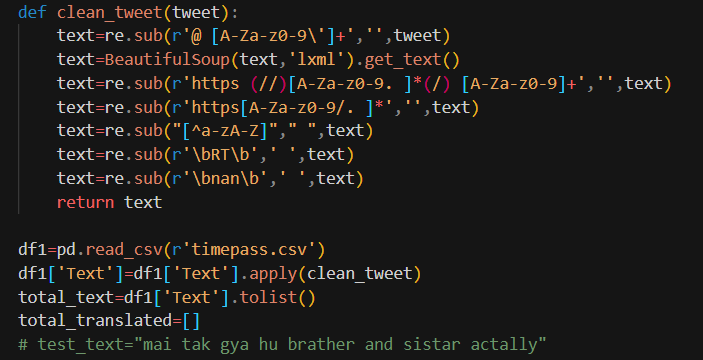
Description automatically generated

Reading the saved vocab

.

#**Note test has been done on the vocab for simple words like good to check if it exists or not**

Duplicates have been removed



Reading and cleaning the timepass.csv which contains the processed test dataset.

Text

Description automatically generated

**NOTE : IT IS WRITTEN english\_vocab but hinglish\_vocab has been read with wrong name so basically work is being done using hinglish\_vocab**

First checking each word in the input text list if it exist in the hinglish vocab

If it does, then its changed with the key of the adjacency value of the found values/

Meaning:

Suppose the sentence is “Mai khul rha hu”

And here khul is found in the adjacency list of khel->khil,**khul**,khol,

So khul would be replaced with khel.

(its just an example not very accurate one as here khul can have different meaning which already tried reducing this kind of interchange in previous steps)

Text

Description automatically generated

Now if the words have not been changed meaning it wasn’t found in the vocab

So here we are finding the levenshtein distance of those words with every word in the vocabulary whose distance is smaller than 2 and then replacing the word with lesser distance

This is done so because it could be that someone has written the word terribly and it wont be in our vocabulary so the correct spelling word which would be nearer to it would be replaced

Text

Description automatically generated

Now the text have been normalized so its being appended to the normalized list

Text

Description automatically generated

Now on the normalised sentence we are doing sentence tagging to identify which words are English and which are **NOT\_ENGLISH**.

NOT ENGLISH because the transformer used for this purpose has good accuracy with tagging the words as English so we only need to distinctify the words in the sentence if its English or not.

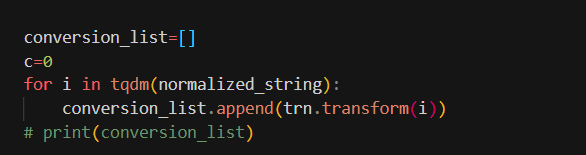
If it wont be English then it sure would be hinglish.

Now in the classify list we are translating the English words to hindi

So the output would be now as :

Normalised string : “mai khel ne ja rha hu you coming?”

CHANGED STRING : “mai khel ne ja rha hu तुम आ रहा”



Now we are transliterating the whole string to hindi devnagri

So

CHANGED STRING WOULD BE NOW :

“मैं खेलने जा राह| हू तुम आ रहा”

Text

Description automatically generated

Finally the whole sentence is now translated to English :

Converted String:

**I am going to play are you coming**

And saved to a file.

2nd:

Text

Description automatically generated

**NOTE Same as above but here English\_vocab is English vocab only.**

Text

Description automatically generated

Now considering unchaned part as hinglish part of string,

For example :

Sentence : Mai khelne ja rha hu you comng?

Changed : Mai khelne ja rha hu you **coming**?

Here coming is removed from the list as its already changed from English vocab meaning its would be an English word.

Text

Description automatically generated

Now going through all the left out words in hinglish vocab and changing the exact match in the vocab

Text

Description automatically generated

Making two separate lists…for hinglish and English parts

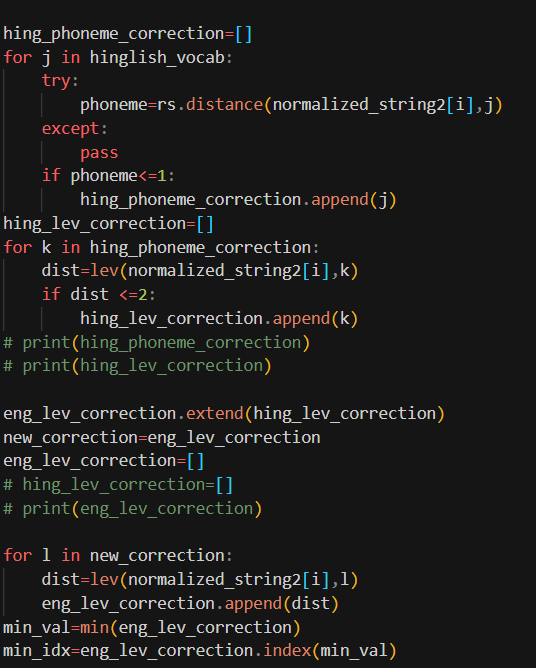
Text

Description automatically generated

**NOW USING PHONEME TO FIND THE SIMILAR SOUNDING WORDS IN THE VOCAB FOR THE ENGLISH UNCHANGED WORDS**

**AND ON THE FOUND WORDS FOR SIMILAR SOUNDS WE ARE CHECKING LEVENSHTEIN DISTANCE FOR WORDS DISTANCE SMALLER OR EQUAL TO 1**

**AND REPLACING THE UNCHANGED WORD WITH THE THE FIRST WORD THAT SATISFY ALL THESE CONDITIONS.**

****

**NOW DOING SAME FOR HINGLISH UNCHANGED WORDS**

**AND ADDING BOTH THE SENTENCES TOGETHER IN ONE LIST**

**Text

Description automatically generated**

**NOW USING PYENCHANT TO FIND SUGGESTIONS AND REPLACING THE STILL UNCHANGED WORDS AFTER ABOVE HINGLISH PARTWISE TWO STEP PROCESS ALSO.**

**Text

Description automatically generated**

now classifying the final normalized string with classifier transformer as English or not

Text

Description automatically generated

Again translating then transliterating and further translating

As done in the 1st method

And saving the final translation into translated file.

.

NOTE ALL THE TRANSLATION DONE BY METHOD 1 IS SAVED INTO :

**total\_translated\_test\_1st\_method**

AND BY METHOD 2 INTO :

**total\_translated2\_test\_2nd\_method**

**SINCE RUNNING TRANSFORMERS AND FINDING WORDS INTO THE VOCAB WAS NOT EASY TASK FOR LOCAL MACHINE SO WE CREATED DATASETS ON KAGGLE AND DONE PROCESSIN OF THE FILES ON KAGGLE FOR 7-10 DAYS.**

**LINKS OF KAGGLE WORK ARE HERE :**

**DATASETS:**

[**https://www.kaggle.com/datasets/tihsrahly/real-translated-3k-2nd-version**](https://www.kaggle.com/datasets/tihsrahly/real-translated-3k-2nd-version)

[**https://www.kaggle.com/datasets/tihsrahly/updated-vocab**](https://www.kaggle.com/datasets/tihsrahly/updated-vocab)

[**https://www.kaggle.com/datasets/tihsrahly/real-translated-3k**](https://www.kaggle.com/datasets/tihsrahly/real-translated-3k)

[**https://www.kaggle.com/datasets/tihsrahly/translated-3k**](https://www.kaggle.com/datasets/tihsrahly/translated-3k)

[**https://www.kaggle.com/datasets/tihsrahly/english-hinglish-erroneous**](https://www.kaggle.com/datasets/tihsrahly/english-hinglish-erroneous)

[**https://www.kaggle.com/datasets/tihsrahly/englishhinglish-vocab**](https://www.kaggle.com/datasets/tihsrahly/englishhinglish-vocab)

[**https://www.kaggle.com/datasets/tihsrahly/oxford-dictionary**](https://www.kaggle.com/datasets/tihsrahly/oxford-dictionary)