# Development & Study of Hinglish to English Translation and Classification Techniques

**Final Year**
**Project Report**

***Submitted by***

**Nicole D'Souza (A020)**
**Devarsh Patel (A048)**
**Jigyashu Saravta (A061)**

***Under The Guidance Of***

**Prof. Ashwini Rao**

***In fulfillment for the award of the degree of***

**B.TECH.**

**INFORMATION TECHNOLOGY**

At

SVKM'S NMIMS Deemed to be UNIVERSITY | MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING

Department of Information Technology
Mukesh Patel School of Technology Management & Engineering
NMIMS (Deemed –to-be University)
JVPD Scheme Bhaktivedanta Swami Marg,
Ville Parle (W), Mumbai-400 056.

**April, 2023**

# CERTIFICATE

This is to certify that the project entitled "**Development & Study of Hinglish to English Translation and Classification Techniques**" is the bonafide work carried out by **Nicole D'Souza, Devarsh Patel, Jigyashu Saravta** of B.Tech (IT), MPSTME, Mumbai, during the VIII Semester of the academic year 2022-2023, in partial fulfillment of the requirements for the award of the degree of Bachelors of Technology as per norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

| | |
|---|---|
| (Signature of Internal Mentor 1) | (Signature of External Examiner) |
| *Name:* | *Name:* |
| *Designation:* | *Designation:* |

| | |
|---|---|
| HOD (IT) | Dean |
| (Dr.Ketan Shah) | (Dr. Alka Mahajan) |

# DECLARATION

We, **Nicole D’Souza, Devarsh Patel, Jigyashu Saravta**, roll numbers: **A020, A048, A061** respectively, understand that plagiarism is defined as any one or the combination of the following:

1. Uncredited verbatim copying of individual sentences, paragraphs or illustrations (such as graphs, diagrams, etc.) from any source, published or unpublished, including the Internet.
2. Uncredited improper paraphrasing of pages or paragraphs (changing a few words or phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did or wrote what. (Source: IEEE, The Institute, Dec. 2004)

We have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of our work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

We affirm that no portion of our work can be considered as plagiarism and we take full responsibility if such a complaint occurs. We understand fully well that the guide of the seminar/project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature:
Name: Nicole D’Souza
Roll No.: A020
Date:

Signature:
Name: Devarsh Patel
Roll No.: A048
Date:

Signature:
Name: Jigyashu Saravta
Roll No.: A061
Date:

# ACKNOWLEDGEMENTS

We would like to take this opportunity to express our heartfelt gratitude to everyone who has contributed to the successful completion of this project. Without their support and guidance, this project would not have been possible.

Firstly, we would like to thank our project supervisor, Dr. Aswhini Rao, for her invaluable guidance, support, and encouragement throughout the project. Her expert knowledge, constructive feedback, and insights were instrumental in shaping the project and helping me stay on track.

We would also like to express our sincere appreciation to NMIMS's MPSTME, for providing us with the necessary resources and facilities to conduct our research. Their timely support and cooperation were crucial in completing this project on time.

We would also like to thank our colleagues and friends, who have provided us with their valuable feedback, suggestions, and motivation throughout the project. Their constant encouragement and support were a source of inspiration for me.

Finally, we'd want to express our appreciation to our family for their continuous support, patience, and understanding during this effort. Throughout the difficult times, their love, encouragement, and moral support have been my driving force.

Again, we would want to offer our deepest appreciation to everyone who has contributed to this project; it would not have been possible without their support and direction.

# TABLE OF CONTENTS

## List of Figures

## List of Tables

## Abstract

Code-mixing is the practice of utilizing multiple languages in a single statement. Code-mixing is a widespread occurrence in multilingual communities all over the world, and it is particularly very prevalent in texts on social media. As a result of the widespread usage of social media programmes, a substantial amount of unstructured text is produced. A rapidly developing area of research in the field of text mining is code-mixing. Due to their modern yet regionalized speech patterns, the current social media posts, blogs, and reviews frequently use code-mixed messaging. There are many distinct uses for linguistic codes from different languages. Hinglish, or code-mixed Hindi and English, is a frequent occurrence in everyday language use in India. Thus, a translation mechanism is required to aid monolingual users and to facilitate language processing models' understanding. The idea of machine translation is put to a new test by this linguistic blending. In this paper, we try to study the different effective strategies for classification and translation and find the gap in the current research work by identifying some of the challenges to machine translation. After comparing and contrasting the two effective existing methodologies for machine translation, we came up with our framework which showed an improvement over the previous methods. However, it did not outperform Google Translate API. We then discussed the various challenges faced by us.

# I. Overview

## 1. Project Specification

### 1.1. Introduction

India is a linguistically diverse nation, that along with a long history of international associations has resulted in a bilingual society where most individuals prefer to speak in a mixed language during informal conversations, the most popular of these code-mixed languages in India is Hinglish, it is a combination of Hindi and English, that is mostly used in social media conversations, thus giving researches a large corpus of unstructured Hinglish text, trying to manually translate this text is a burdensome task, therefore we need to rely on pre-processing techniques to manage this data. Most NLP based research has been done on the English language and there isn't a lot of information on how to effectively analyse other languages like Hindi, Marathi, Chinese, Japanese etc. A code-mixed language like Hinglish simply adds an extra layer of complexity. Through our final year project, we aim to tackle these issues by identifying the current tools and techniques used by researchers on how to manage Hinglish text, identify the research gaps and inadequacies in these techniques and propose a new framework on how to identify Hinglish text and effectively translate it to English.To appropriately identify the research gaps we first implement the frameworks described in two research papers, the first paper suggests using google translate to individually translate hinglish words and merge them with the rest of the english sentence, in order to test it's effectiveness we used google translate to translate the entire sentence, our models showed the proposed model to have a higher BLEU score then google translate.

The second paper that we implemented, first trained a model to identify which words are code mixed hindi and which are english,the tokens that were identified as hinglish were then first transliterated to devnagri hindi and then to english, in the future we aim to fine tune our existing model further by replacing the google translate api with google neural machine translation or mbart (a sequence to sequence model fine tuned to translate hinglish to english).

### 1.2. Importance

Code Mixing is a common phenomenon found in multilingual societies, in India code mixing is done through transliteration and randomly blending Hindi and English, thus creating Hinglish. Most Indian users are found to communicate through Hinglish across various social media platforms, this results in a large amount of data that can't be managed due to the lack of appropriate tools and techniques, the aim of this project is to survey existing tools and techniques on managing hinglish data, identify their inadequacies and work towards building a model that effectively classifies text as either Hinglish or English and translates the Hinglish text to its English equivalent. The identification and translation of Hinglish text will enable future researchers to build models that can classify or analyse Hinglish texts more effectively.

### 1.3. Objective & Scope

Code Mixing is a common phenomenon found in multilingual societies, in India code mixing is done through transliteration and randomly blending Hindi and English, thus creating Hinglish. Most Indian users are found to communicate through Hinglish across various social media platforms, this results in a large amount of data that can't be managed due to the lack of appropriate tools and techniques, the aim of this project is to survey existing tools and techniques on managing hinglish data, identify their inadequacies and work towards building a model that effectively classifies text as either Hinglish or English and translates the Hinglish text to its English equivalent. The identification and translation of Hinglish text will enable future researchers to build models that can classify or analyse Hinglish texts more effectively.

## 1.4. Deliverables

- Implementation of two existing techniques used for Hinglish to English text classification.
- Research Paper on the study and implementation of Hinglish to English Classification & Translation.

## 2. Literature Survey

Indian English as defined by one of the most revered scholars of Indian English is an institutionalized second-language variety of English that has a long history of acculturation and a large range of functions in the local, educational, administrative, and legal systems which has developed nativized discourse and style types and functionally determined sublanguages.He attributes code mixing to a sociolinguistic consequence of colonialism,this phenomenon is commonly seen in several countries Nigeria, China, Bangladesh, Japan etc [1].

In India code mixing is done through transliteration and randomly blending Hindi and English, thus creating Hinglish.Most Indian users are found to communicate through Hinglish across various social media platforms,this becomes a problem for researchers because Hinglish doesn't follow fixed standards for spelling and grammar users simply employ the phonetics of a word to come up with spelling e.g no in hindi translate to *नहीं* which in hinglish can be written as *nahi,nhi,nai* etc based on a variety of reasons like regional pronunciations or dialectal conventions[2].

Moreover, unlike monolingual languages, there is no formal data in the form of news articles or books. This results in researchers having to resort to using unstructured data from social media comments and posts [4]. Several researches have worked on translating hinglish code mixed text to english, in [3] the researchers first look for idioms or phrase in the sentences extract them and translate them as a whole to do this they used a dataset of over 12,000 rows, the rest of the sentences are tokenized and tagged as hindi, english or other, these tokens are analysed morphologically and reverse morphologically after which pos tagging is done, these words are then arranged and translated.The researchers found that the sentences that were translated into hindi had a much higher accuracy than those translated to english as hinglish is based on hindi grammar.The authors of [4] fine tuned a mBART, which is a multilingual sequence-to-sequence denoising auto-encoder. It has been pre-trained using the BART objective on large-scale monolingual corpora of 25 languages extracted from Common Crawl, this mBART has been fine tuned using the precog dataset, this fine tuned mBART is called mBART-hien-cm and has a BLEU score of 33.30.In [5] existing translating services like google translate and bing translate are tested out after which the proposed pipeline is developed in which the sentence is tokenized and only the hindi tokens are translated to english using google translate api, this method obtained a BLEU score of 0.153 compared to the 0.14 obtained on google translate. The authors of [6] trained a 2 layer LSTM model to identify between hinglish and english words,they then transliterated these hinglish words to devnagri hindi and translated them to english their model obtained a BLEU score of 0.496. In [7] a similar approach is followed however instead of using google translate to perform the translation they make use of Google Neural Machine Translator.In [9], the authors also fine tuned a mBART model along with the Google Neural Machine Translation instead of translating the text to english they first translated it to hindi and then to english, this model was found to have a BLeU score of 15.3.The authors of [10] performed a comparative analysis between PHINC,mBART and mt5 they found that mt5 provided the highest BLeU score.

In this report we have implemented the methodologies described in [5] and [6] in the future we hope to build upon their work by employing GNMT or mBART instead of Google translate

# II. Analysis & Design

## 1. Requirement Analysis

- **Data Collection:**
  We made use of the popular dataset released by [8],the dataset consists of 13,760 rows containing social media comments, tweets in hinglish that have been translated to their english equivalent.For the model we use a precog dataset consisting of 12,000 words of hinglish and english words each labeled as hindi or english.

- **Preprocessing:**
  ML Model: For the model the hi/en labels were encoded using a label encoder into 0 & 1, the words were converted to lowercase and then vectorized Special Characters, Links: All special characters and numbers were taken out of the dataset,all links in the dataset were also deleted using regex and the sentences were converted to lowercase.

- **Machine Learning Model:**
  The model used was a 2 layer LSTM with relu and sigmoid activation functions and an adam optimizer.The model obtained an accuracy of 90% .

- **Machine Transliteration:**
  We transliterated the text from hinglish to devanagari hindi with the help of Indic Transliteration and hindi_wsd package.

- **Grammar & Spell Correction:**
  As the transliterated data contained several spelling and grammar errors, in order to correct the spelling errors we made use of the Spello model,to fix the grammar errors we made use of Gramformer.

- **Machine Translation:**
  Since hinglish follows the conventions of hindi grammar rather than english grammar, the english tokens were translated to devnagri hindi using google translate. All the tokens were then merged together and then translated to english using google translate.

- **Evaluation:**
  The translated output was compared with the original english sentences and evaluated using BLeU score. BLEU (BiLingual Evaluation Understudy) is a metric for automatically evaluating machine-translated text. The BLEU score is a number between zero and one that measures the similarity of the machine-translated text to a set of high quality reference translations.

## 2. Feasibility Study

- **Data Availability:**
  The availability of a suitable dataset is critical to the success of the project. A diverse and comprehensive dataset of code-mixed hinglish along with accurate translations is essential to gauging the efficacy of the frameworks proposed.Moreover a dataset consisting of hinglish words and their equivalent english translations can help train the classification model.The availability of such a dataset should be analyzed to determine if it is feasible to collect or obtain one. Several authors have released their datasets for public use; the cumulative size of such a dataset is approximately 30,000 rows.

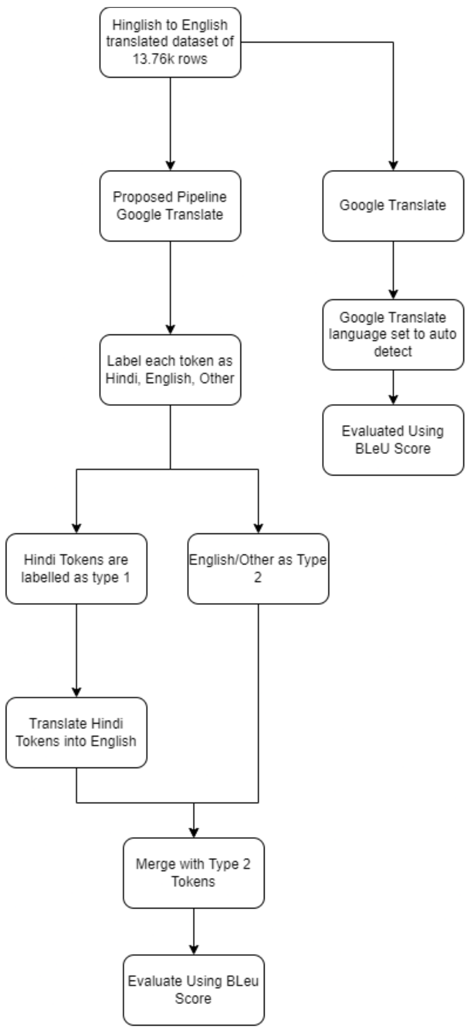
- **Resource Availability:**
  The development of a code-mixed language classification and translation model requires computing resources such an internet connections,cloud storage and GPUs. The availability of such resources should be analyzed to determine if they are feasible to obtain or allocate. Google's collab environment, or kaggle's Notebooks will be used to create this project with the available GPU being sufficient for this project's training.

- **Model Performance:**
  The performance of the machine learning model should be analyzed to determine if it is feasible to achieve acceptable levels of accuracy,precision, recall, and F1 score for the detection of code-mixed language.The quality of the translations is verified using the BLeU score.

Considering all the above mentioned categories, we conclude that the project is feasible and can be conducted in the given timeline with the available resources.

## 3. Design Development.

### 3.1. PHINC: A Parallel Hinglish Social Media Code-Mixed Corpus for Machine Translation



Hinglish to English translated dataset of 13.76k rows
Proposed Pipeline Google Translate
Google Translate
Google Translate language set to auto detect
Evaluated Using BLeU Score
Label each token as Hindi, English, Other
Hindi Tokens are labelled as type 1
English/Other as Type 2
Translate Hindi Tokens into English
Merge with Type 2 Tokens
Evaluate Using BLeu Score

Fig. 1 : PHNC Flowchart

- **Dataset:**
  We made use of the popular dataset released by [8],the dataset consists of 13,760 rows containing social media comments, tweets in hinglish that have been translated to their english equivalent.

- **Preprocessing:**
  All special characters and numbers were taken out of the dataset,all links in the dataset were also deleted using regex and the sentences were converted to lowercase.The text was then tokenized each token was marked as hindi,english or other using google detect.

- **Google translate:**
  In this method the source language was set to auto detect and the destination was set to english. The corpus was then evaluated using BLeU scores for unigrams,bigrams,trigrams and n grams.

- **Proposed Pipeline with Google Translate:**
  Here the tokens that were marked as Hindi were translated to English using the google translate api and then merged with the rest of the tokens to form full sentences.This too was evaluated using BLeU scores.

### 3.2. Code-Mixed Hinglish to English Language Translation Framework

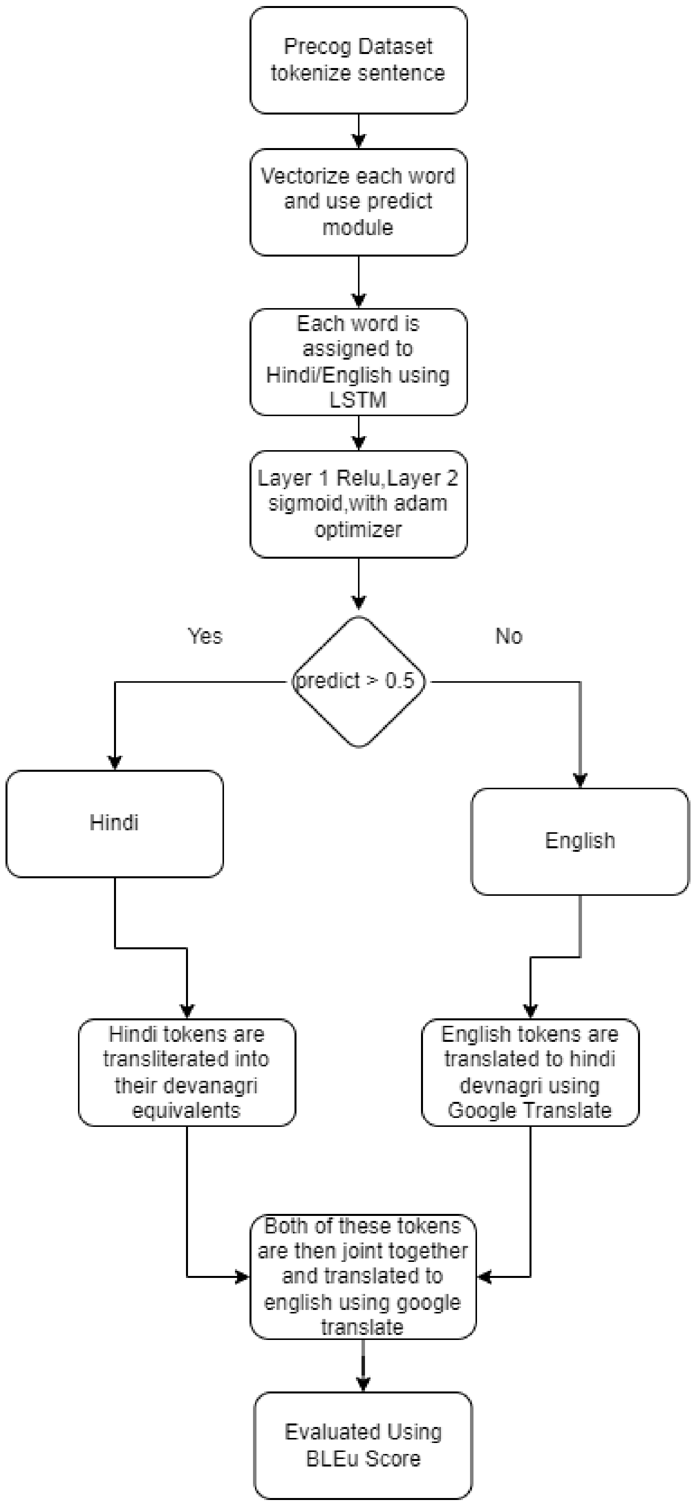

Precog Dataset tokenize sentence
Vectorize each word and use predict module
Each word is assigned to Hindi/English using LSTM
Layer 1 Relu,Layer 2 sigmoid,with adam optimizer
Yes
predict > 0.5
No
Hindi
English
Hindi tokens are transliterated into their devanagri equivalents
English tokens are translated to hindi devnagri using Google Translate
Both of these tokens are then joint together and translated to english using google translate
Evaluated Using BLEu Score

Fig. 2: CMHE flowchart

- **Datasets**:
  For the model we use a precog dataset consisting of 12,000 words of hinglish and english words each labeled as hindi or english. For the machine transliteration & translation we used the dataset released by [8],the dataset consists of 13,760 rows containing social media comments, tweets in hinglish that have been translated to their english equivalent.
- **Preprocessing:**

ML Model : for the model the hi/en labels were encoded using a label encoder into 0 & 1, the words were converted to lowercase and then vectorized.
Machine Transliteration & Translation : All special characters and numbers were taken out of the dataset,all links in the dataset were also deleted using regex and the sentences were converted to lowercase.

- **Machine Learning Model:**
  The model used was a 2 layer LSTM with relu and sigmoid activation functions and an adam optimizer.The model obtained an accuracy of 90%.

- **Machine Transliteration**:
  We transliterated the text from hinglish to devanagari hindi with the help of Indic Transliteration and hindi_wsd package, this transliteration did not follow proper spelling and grammar rules of hindi so to perform a spell & grammar correction we made use of the spello package.

- **Machine Translation**:
  Since hinglish follows the conventions of hindi grammar rather than english grammar, the english tokens were translated to devnagri hindi using google translate. All the tokens were then merged together and then translated to english using google translate.The translations were then evaluated using the BLeU score.

## 3.3. Proposed Hinglish to English Classification & Language Translation Framework

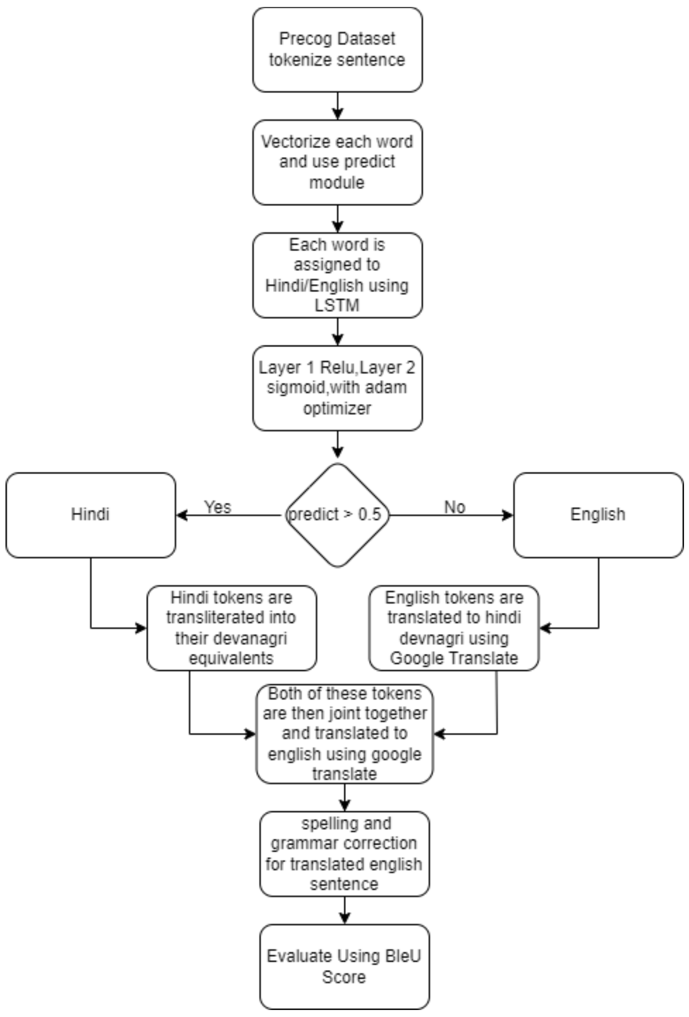

Precog Dataset tokenize sentence
Vectorize each word and use predict module
Each word is assigned to Hindi/English using LSTM
Layer 1 Relu,Layer 2 sigmoid,with adam optimizer
predict > 0.5
Yes
No
Hindi
English
Hindi tokens are transliterated into their devanagri equivalents
English tokens are translated to hindi devnagri using Google Translate
Both of these tokens are then joint together and translated to english using google translate
spelling and grammar correction for translated english sentence
Evaluate Using BleU Score

Fig. 3: Proposed framework flowchart

We found out that a lot of sentences after getting translated having spelling and grammatical errors. We thought of adding an extra layer for spelling and grammar correction to our architecture in order to improve the accuracy.

The devnagri hindi translations and transliterations featured numerous misspellings and grammatical errors. Hence, the translations received a low BLeU score. In order to address this issue, the Spello model was employed to correct the spellings of the translated Devnagri Hindi words. This model uses phonetic principles to identify words with comparable sounds.

Example:

Incorrect Spelling 1 : अच्चा

Correct Spelling 2 : अच्छा

Gramformer is a model that identifies grammatical errors in sentences and replaces them with their grammatically correct alternatives in order to address the issue of bad sentence grammar.

Example:
Incorrect Sentence 1 : We likes Pizza
Correct Sentence 1 : We like Pizza

Incorrect Sentence 2 : How is you ?
Correct Sentence 2 : How are you ?

## 4. Technology and Software Details

- **COLAB Notebook**

  Google Colab is a popular online platform for data science and machine learning that allows users to run Python code and access powerful computing resources, such as GPUs and TPUs, for free.

  Google Colab provides access to powerful computing resources that are needed for training and running machine learning models, including those for Hinglish to English translation. Users can import and preprocess large datasets, train and fine-tune deep learning models, and evaluate the performance of their models using metrics such as accuracy, precision, and recall.

  In summary, Google Colab provides a convenient and powerful platform for building and training Hinglish to English translation models, making it an essential tool for researchers and practitioners in the field of natural language processing.

- **Sklearn**

  Scikit-learn (sklearn) is a popular Python library for machine learning. However, it is not specifically designed for language translation tasks such as Hinglish to English translation.

  Language translation typically involves complex natural language processing (NLP) techniques, such as machine translation and sequence-to-sequence modeling, which are not directly supported by sklearn. That being said, sklearn can be used in conjunction with other NLP libraries and frameworks to perform tasks such as text preprocessing, feature extraction, and model training. For example, sklearn provides a number of feature extraction techniques, such as CountVectorizer and TfidfVectorizer, which can be used to convert text data into numerical features that can be used as inputs to machine learning models.

  In the context of Hinglish to English translation, sklearn could potentially be used to preprocess and transform the input and output data into suitable numerical representations, which could then be used to train a machine translation model using other frameworks or libraries such as TensorFlow or PyTorch. However, it is important to note that machine translation is a challenging task, and achieving high-quality translations requires careful consideration of various factors, such as data quality, model architecture, and training strategies.

- **Python**

  Python can play an important role in Hinglish to English translation by using natural language processing (NLP) techniques. Hinglish is a blend of Hindi and English languages and is commonly used in India. Here are some ways Python can be used for Hinglish to English translation:

  Preprocessing: The first step in Hinglish to English translation is to preprocess the text. Python can be used to remove unnecessary characters, tokenize the text into individual words, and remove stopwords (commonly used words in a language that do not carry significant meaning).

  Language detection: Python can be used to detect the language of the text. This is important because the translation process will differ based on the input language.

Translation: Once the language of the input text is detected, Python can be used to translate the Hinglish text to English using machine translation libraries like Google Translate, Microsoft Translator, or Yandex Translate. These libraries use statistical or neural machine translation techniques to provide accurate translations.

Post-processing: After translation, the output text can be post-processed to ensure that the translated text is grammatically correct and reads well in English.

- **Spacy**
  Spacy is a popular open-source library for natural language processing (NLP) in Python.Spacy is a valuable tool in the translation of Hinglish to English, particularly for tasks such as text preprocessing. However, for more complex translation tasks, it may be necessary to use a specialized machine translation tool that is specifically designed for handling Hinglish.

- **NLTK**
  NLTK (Natural Language Toolkit) is a powerful Python library that is widely used for natural language processing (NLP) tasks, including machine translation. However, when it comes to Hinglish to English translation, which involves a mix of Hindi and English languages, NLTK alone may not be sufficient.

  To translate Hinglish to English, one approach is to use a combination of NLP techniques and machine learning algorithms. NLTK can be used to preprocess the text by tokenizing the words, identifying parts of speech, and performing other basic NLP tasks.

- **Text Blob**
  In the context of Hinglish to English translation, TextBlob can be used to identify the parts of the text that are in English and those that are in Hindi (or Hinglish, which is a mix of Hindi and English). This is done using TextBlob's language detection feature, which can automatically detect the language of a piece of text.

- **Gramformer**
  Gramformer is a neural language model that is capable of generating high-quality text by correcting grammatical errors, rephrasing sentences, and making other improvements to the language. In the context of Hinglish to English translation, Gramformer can be used to improve the quality of the translated text by correcting grammatical errors and ensuring that the resulting text reads fluently and naturally.

  For example, suppose the original Hinglish sentence is "Mera ek dost hai jiske paas ek car hai," which translates to "I have a friend who has a car" in English. However, this translation is grammatically incorrect because the word "has" should be replaced with "owns." Gramformer can automatically correct this error, resulting in the correct translation: "I have a friend who owns a car."

  Overall, Gramformer can play an important role in improving the quality of Hinglish to English translations by correcting grammatical errors, rephrasing sentences, and ensuring that the resulting text reads fluently and naturally in English.

- **Indic Transliteration**

Indic Transliteration can play a significant role in Hinglish to English translation by providing a way to represent Hindi or other Indic language words using the Roman script. This allows for a more seamless and efficient translation process, as the transliterated words can be easily recognized and converted into their corresponding English equivalents.

For example, if the Hindi word "खुशी" (khushi) needs to be translated into English, it can be transliterated as "khushi" and then translated into "happiness". This can be particularly useful when working with Hinglish, which is a blend of Hindi and English, as it allows for the Hindi words to be represented in a way that can be easily understood by English speakers.

Transliteration also helps to preserve the pronunciation of the original word, which can be useful for non-native speakers who may not be familiar with the Devanagari script used for Hindi and other Indic languages.

- **Matplotlib**
  Matplotlib is a popular data visualization library in Python that can be used to create graphs, charts, and other visualizations of data. While Matplotlib itself does not have a direct role in Hinglish to English translation, it can be used in conjunction with other Python libraries and tools to create visualizations that help in the translation process.

- **Google Translate**
  Google Translate can be useful for translating text from Hinglish to English, as it has a dedicated translation feature for this language pair. Hinglish is a hybrid language that blends Hindi and English, and Google Translate's algorithms have been trained on large datasets of this language pair, which allows it to produce relatively accurate translations.

- **Spello**
  Spello is a software tool that can be used to help with Hinglish to English translation. Hinglish is a mix of Hindi and English, which can make it difficult for some people to understand or translate. Spello can help by automatically detecting and correcting spelling and grammar errors in the Hinglish text, which can make it easier to translate accurately into English.

## 5. Project Planning

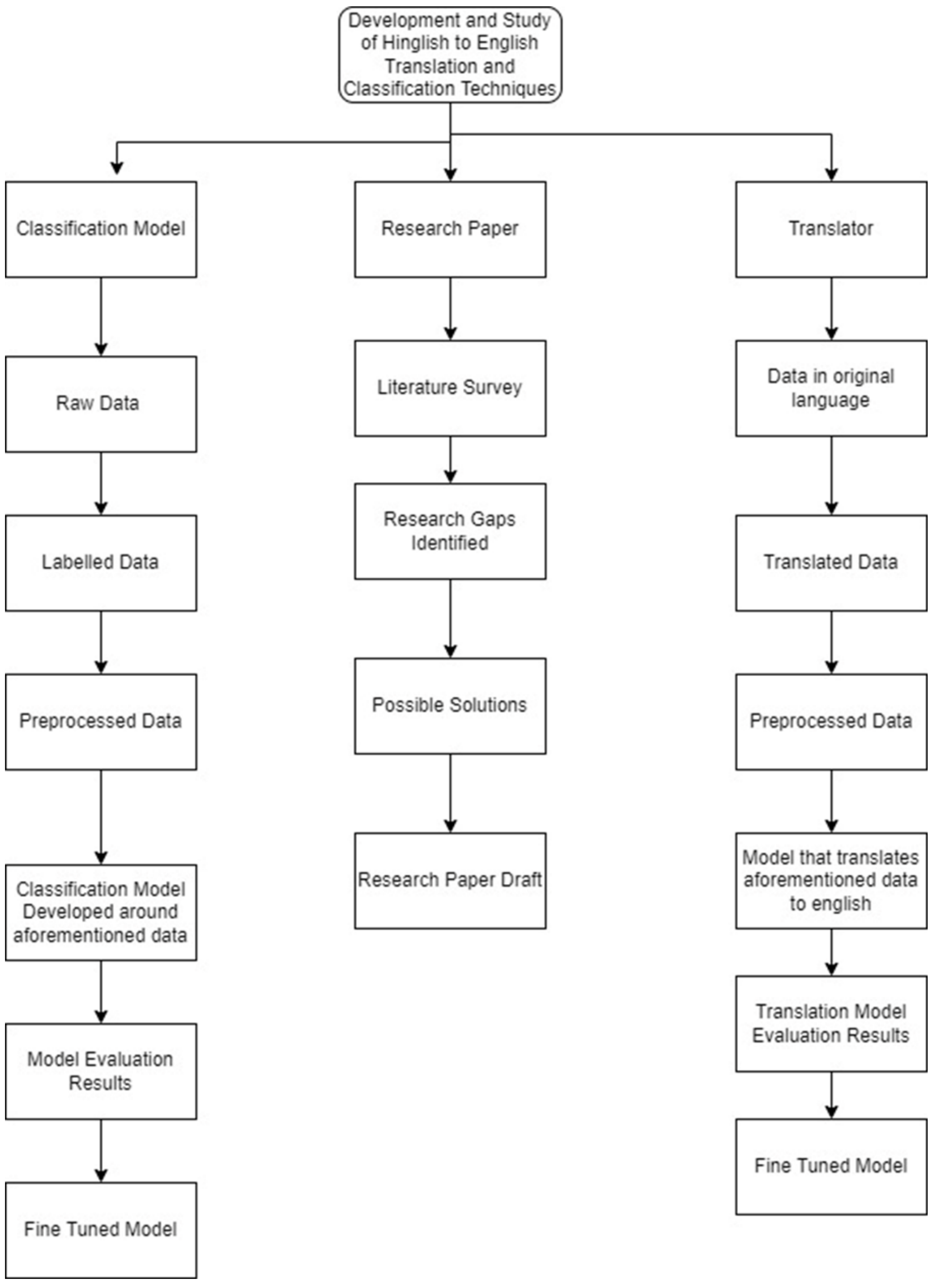

Development and Study of Hinglish to English Translation and Classification Techniques
Classification Model
Raw Data
Labelled Data
Preprocessed Data
Classification Model Developed around aforementioned data
Model Evaluation Results
Fine Tuned Model
Research Paper
Literature Survey
Research Gaps Identified
Possible Solutions
Research Paper Draft
Translator
Data in original language
Translated Data
Preprocessed Data
Model that translates aforementioned data to english
Translation Model Evaluation Results
Fine Tuned Model

Fig. 4: PBS

A work breakdown structure has been developed based on the product breakdown structure. This structure divides the process of developing the product into a number of activities and the subtasks associated with each activity.

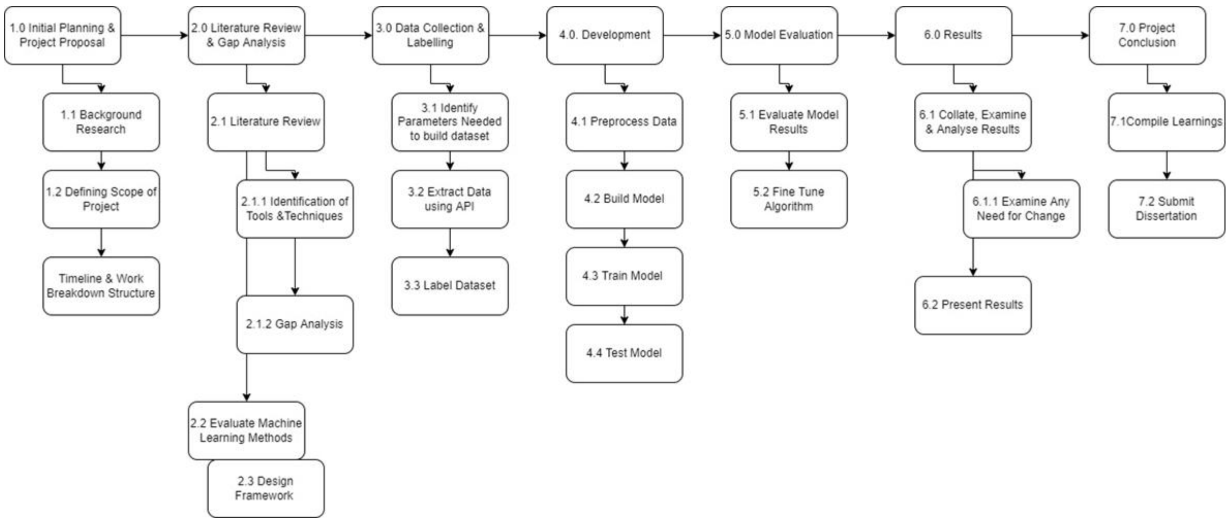
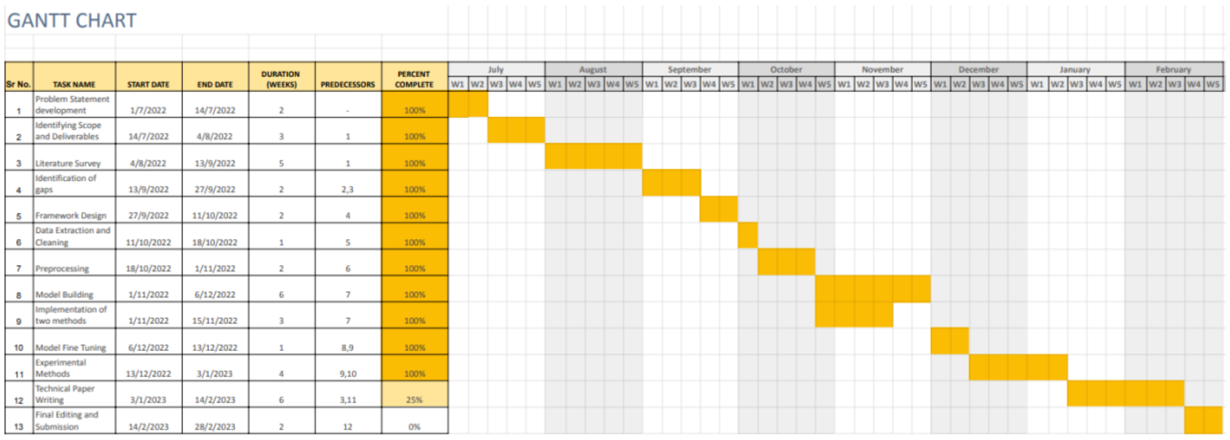

1.0 Initial Planning & Project Proposal
1.1 Background Research
1.2 Defining Scope of Project
Timeline & Work Breakdown Structure
2.0 Literature Review & Gap Analysis
2.1 Literature Review
2.1.1 Identification of Tools &Techniques
2.1.2 Gap Analysis
2.2 Evaluate Machine Learning Methods
2.3 Design Framework
3.0 Data Collection & Labelling
3.1 Identify Parameters Needed to build dataset
3.2 Extract Data using API
3.3 Label Dataset
4.0. Development
4.1 Preprocess Data
4.2 Build Model
4.3 Train Model
4.4 Test Model
5.0 Model Evaluation
5.1 Evaluate Model Results
5.2 Fine Tune Algorithm
6.0 Results
6.1 Collate, Examine & Analyse Results
6.1.1 Examine Any Need for Change
6.2 Present Results
7.0 Project Conclusion
7.1Compile Learnings
7.2 Submit Dissertation

Fig. 5: WBS

We have created a comprehensive Gantt chart as part of our chronological action plan. This chart takes into consideration all of the necessary actions outlined in the WBS, the amount of time needed to complete each one, and a buffer period of time in case there are any delays that cannot be avoided. From this, an estimate of the project's entire length can also be derived, and it falls well within the parameters of the deadlines.


GANTT CHART

| Sr No. | TASK NAME | START DATE | END DATE | DURATION (WEEKS) | PREDECESSORS | PERCENT COMPLETE |
|---|---|---|---|---|---|---|
| 1 | Problem Statement development | 1/7/2022 | 14/7/2022 | 2 | - | 100% |
| 2 | Identifying Scope and Deliverables | 14/7/2022 | 4/8/2022 | 3 | 1 | 100% |
| 3 | Literature Survey | 4/8/2022 | 13/9/2022 | 5 | 1 | 100% |
| 4 | Identification of gaps | 13/9/2022 | 27/9/2022 | 2 | 2,3 | 100% |
| 5 | Framework Design | 27/9/2022 | 11/10/2022 | 2 | 4 | 100% |
| 6 | Data Extraction and Cleaning | 11/10/2022 | 18/10/2022 | 1 | 5 | 100% |
| 7 | Preprocessing | 18/10/2022 | 1/11/2022 | 2 | 6 | 100% |
| 8 | Model Building | 1/11/2022 | 6/12/2022 | 6 | 7 | 100% |
| 9 | Implementation of two methods | 1/11/2022 | 15/11/2022 | 3 | 7 | 100% |
| 10 | Model Fine Tuning | 6/12/2022 | 13/12/2022 | 1 | 8,9 | 100% |
| 11 | Experimental Methods | 13/12/2022 | 3/1/2023 | 4 | 9,10 | 100% |
| 12 | Technical Paper Writing | 3/1/2023 | 14/2/2023 | 6 | 3,11 | 25% |
| 13 | Final Editing and Submission | 14/2/2023 | 28/2/2023 | 2 | 12 | 0% |

Fig. 6: Gantt Chart

| SN | Task | Duration in weeks | Start Date | End Date | Predecessors |
|---|---|---|---|---|---|
| 1 | Problem Statement development | 2 | 01/07/22 | 14/07/22 | - |
| 2 | Identifying Scope and Deliverables | 3 | 14/07/22 | 04/08/22 | 1 |
| 3 | Literature Survey | 5 | 04/08/22 | 13/09/22 | 1 |
| 4 | Identification of gaps | 2 | 13/09/22 | 27/09/22 | 2,3 |
| 5 | Framework Design | 2 | 27/09/22 | 11/10/22 | 4 |
| 6 | Data Extraction and Cleaning | 1 | 11/10/22 | 18/10/22 | 5 |
| 7 | Preprocessing | 2 | 18/10/22 | 01/11/22 | 6 |
| 8 | Model Building | 6 | 01/11/22 | 06/12/22 | 7 |
| 9 | Implementation of two methods | 3 | 01/11/22 | 15/11/22 | 7 |
| 10 | Model Fine Tuning | 1 | 06/12/22 | 13/12/22 | 8,9 |
| 11 | Experimental Methods | 4 | 13/12/22 | 03/01/23 | 9,10 |
| 12 | Technical Paper Writing | 6 | 03/01/23 | 14/02/23 | 3,11 |
| 13 | Final Editing | 2 | 14/02/23 | 28/02/23 | 12 |

Table 1: Gantt Chart

## III. Project Description

### 1. Workflow

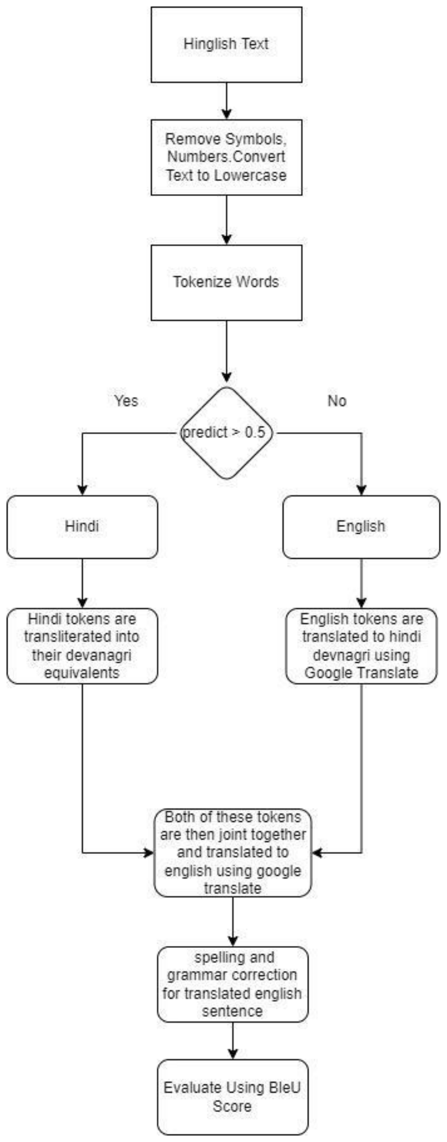

Hinglish Text
Remove Symbols, Numbers.Convert Text to Lowercase
Tokenize Words
Yes
predict > 0.5
No
Hindi
English
Hindi tokens are transliterated into their devanagri equivalents
English tokens are translated to hindi devnagri using Google Translate
Both of these tokens are then joint together and translated to english using google translate
spelling and grammar correction for translated english sentence
Evaluate Using BleU Score

Fig. 7: Project workflow

First the Hinglish text undergoes preprocessing which includes remove special characters,numbers and converting the text to lowercase.The text is then tokenized and passed through the LSTM model which classifies it as Hinglish or English.The tokens classified as Hinglish is transliterated into Devnagri Hindi while the tokens classified as English are translated into Devnagri Hindi.Both these tokens are joint together and translated to English using Google Translate.The translated text then undergoes a spelling and grammar check,after which it is evaluated using BleU score.

## 2. Modules and Components

- **Dataset:**
  For the model we use a precog dataset consisting of 12,000 words of hinglish and english words each labeled as hindi or english. For the machine transliteration & translation we used the dataset released by [8],the dataset consists of 13,760 rows containing social media comments, tweets in hinglish that have been translated to their english equivalent.

- **Preprocessing:**
  All special characters and numbers were taken out of the dataset,all links in the dataset were also deleted using regex and the sentences were converted to lowercase.The text was then tokenized each token was marked as hindi,english or other using google detect.

  ML Model : for the model the hi/en labels were encoded using a label encoder into 0 & 1, the words were converted to lowercase and then vectorized

  Machine Transliteration & Translation : All special characters and numbers were taken out of the dataset,all links in the dataset were also deleted using regex and the sentences were converted to lowercase.

- **Machine Learning Model:**
  The model used was a 2 layer LSTM with relu and sigmoid activation functions and an adam optimizer.The model obtained an accuracy of 90%

- **Machine Transliteration:**
  We transliterated the text from hinglish to devanagari hindi with the help of Indic Transliteration  and hindi_wsd package, this transliteration did not follow proper spelling and grammar rules of hindi so to perform a spell & grammar  correction we made use of the spello package.

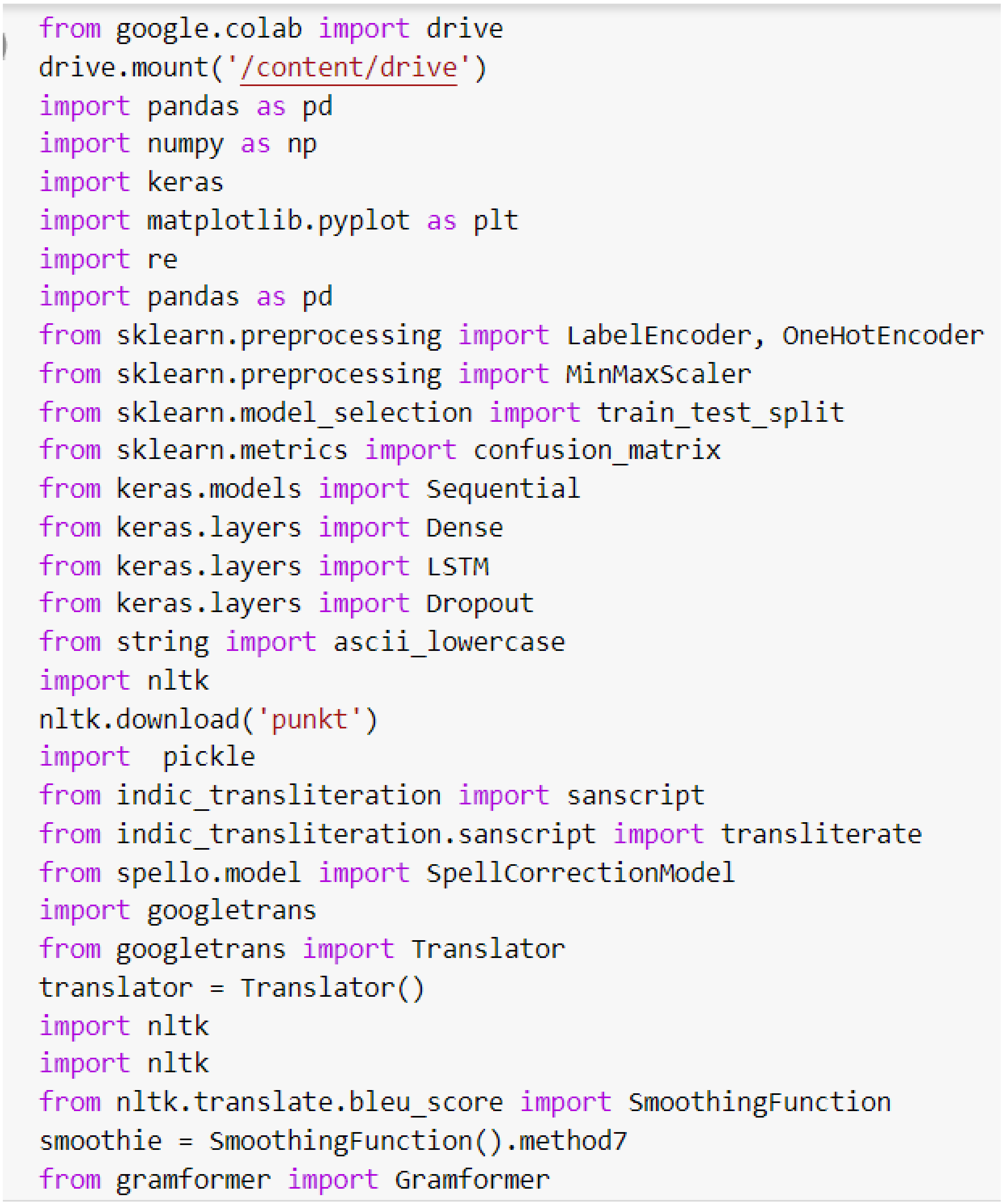
- **Machine Translation:**
  Since hinglish follows the conventions of hindi grammar rather than english grammar, the english tokens were translated to devnagri hindi using google translate. All the tokens were then merged  together and then translated to english using google translate.The translations were then evaluated using the BLeU score.

- **Spell Correction:**
  The devnagri hindi translations and transliterations featured numerous misspellings and grammatical errors. Hence, the translations received a low BLeU score. In order to address this issue, the Spello model was employed to correct the spellings of the translated Devnagri Hindi words. This model uses phonetic principles to identify words with comparable sounds.

- **Grammar Correction:**
  Gramformer is a model that identifies grammatical errors in sentences and replaces them with their grammatically correct alternatives in order to address the issue of bad sentence grammar.

## IV. Project Implementation

1. **Importing Libraries**



```
from google.colab import drive
drive.mount('/content/drive')
import pandas as pd
import numpy as np
import keras
import matplotlib.pyplot as plt
import re
import pandas as pd
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from string import ascii_lowercase
import nltk
nltk.download('punkt')
import  pickle
from indic_transliteration import sanscript
from indic_transliteration.sanscript import transliterate
from spello.model import SpellCorrectionModel
import googletrans
from googletrans import Translator
translator = Translator()
import nltk
import nltk
from nltk.translate.bleu_score import SmoothingFunction
smoothie = SmoothingFunction().method7
from gramformer import Gramformer
```

Screenshot 1: Libraries

- **Pandas**

Pandas is a popular open-source Python data manipulation, analysis, and visualisation package. It has a strong data structure known as DataFrame, which is a two-dimensional table akin to a spreadsheet or a SQL table. The library is based on the NumPy package and provides a variety of data manipulation and data analysis operations, such as merging and combining datasets, filtering and choosing data, grouping data, and more.

- **Numpy**
  NumPy is a Python library for numerical computation. It offers a comprehensive set of mathematical functions and procedures for dealing with big, multi-dimensional arrays and matrices.

- **Keras**
  Keras is a Python-based open-source neural network framework that simplifies the development and training of deep learning models. Keras offers a simple and straightforward interface for designing and training neural networks, as well as a modular and extensible architecture that facilitates customization and experimentation.

- **Matplolib**
  Keras is a Python-based open-source neural network framework that simplifies the development and training of deep learning models. Keras offers a simple and straightforward interface for designing and training neural networks, as well as a modular and extensible architecture that facilitates customization and experimentation.

- **Sklearn**
  Sklearn is a well-known Python machine learning and data mining package. Sklearn provides a uniform API for a broad variety of machine learning techniques, making it simple to test and compare various models. It also provides data preparation methods including feature scaling, feature selection, and data imputation.

- **NLTK**
  The Natural Language Toolkit (NLTK) is a well-known open-source Python framework for natural language processing (NLP). Tokenization, stemming, part-of-speech tagging, parsing, machine learning, and other text processing and analysis functions are included in NLTK. It also provides access to a number of corpora, including as the Brown Corpus,

the Gutenberg Corpus, and the Penn Treebank Corpus, which are frequently used for NLP research and instruction.

- **Pickle**

  Python's pickle library is used to serialise and de-serialize Python objects. The pickle library may be used to retain complicated data structures such as lists, dictionaries, and even bespoke objects on disc. The data that has been saved may subsequently be accessed and utilised by the same or another Python programme.

- **Indic_transliteration**

  Indic_transliteration is a Python package that offers a collection of methods for transliterating text between several Indian subcontinent writing systems such as Devanagari, Tamil, Telugu, Bengali, and others.

- **Spello**

  Spello is a piece of software that may assist in Hinglish to English translation. Hinglish is a blend of Hindi and English that some people find difficult to comprehend or interpret. Spello can assist by automatically identifying and fixing spelling and grammatical mistakes in the Hinglish text, making it simpler to correctly translate into English.

- **Googletrans**

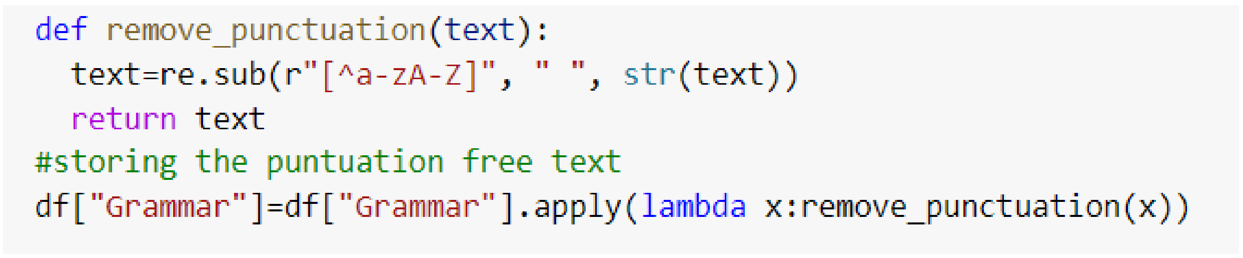
  GoogleTrans is a Python package that offers access to the Google Translate API. Using Python code, you can simply convert text from one language to another. The library supports a broad variety of languages and can handle massive quantities of text translation.

- **Gramformer**

  Gramformer is a Python grammar correction and text creation package. It is based on the cutting-edge language model GPT-3 and includes a user-friendly interface for fine-tuning the model for particular purposes such as text correction, paraphrasing, and text synthesis.
- **Torch**

  Torch is an open-source machine learning package that is mainly used to construct and train deep learning models. Torch provides a variety of tools for creating and training neural networks, such as different kinds of layers, loss functions, and optimisation methods. It also includes data loading and preparation modules, as well as tools for visualising and monitoring model performance.

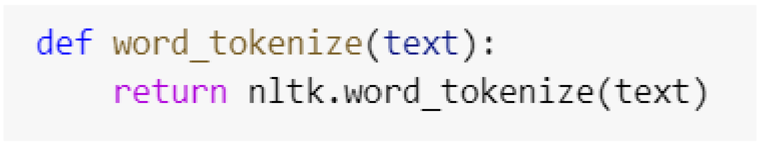
## 2. Preprocessing



```
def remove_punctuation(text):
  text=re.sub(r"[^a-zA-Z]", " ", str(text))
  return text
#storing the puntuation free text
df["Grammar"]=df["Grammar"].apply(lambda x:remove_punctuation(x))
```

Screenshot 2: Removing punctuation

First step of pre-processing is to remove punctuation,hashtags and other characters,a function to do that is as defined above.

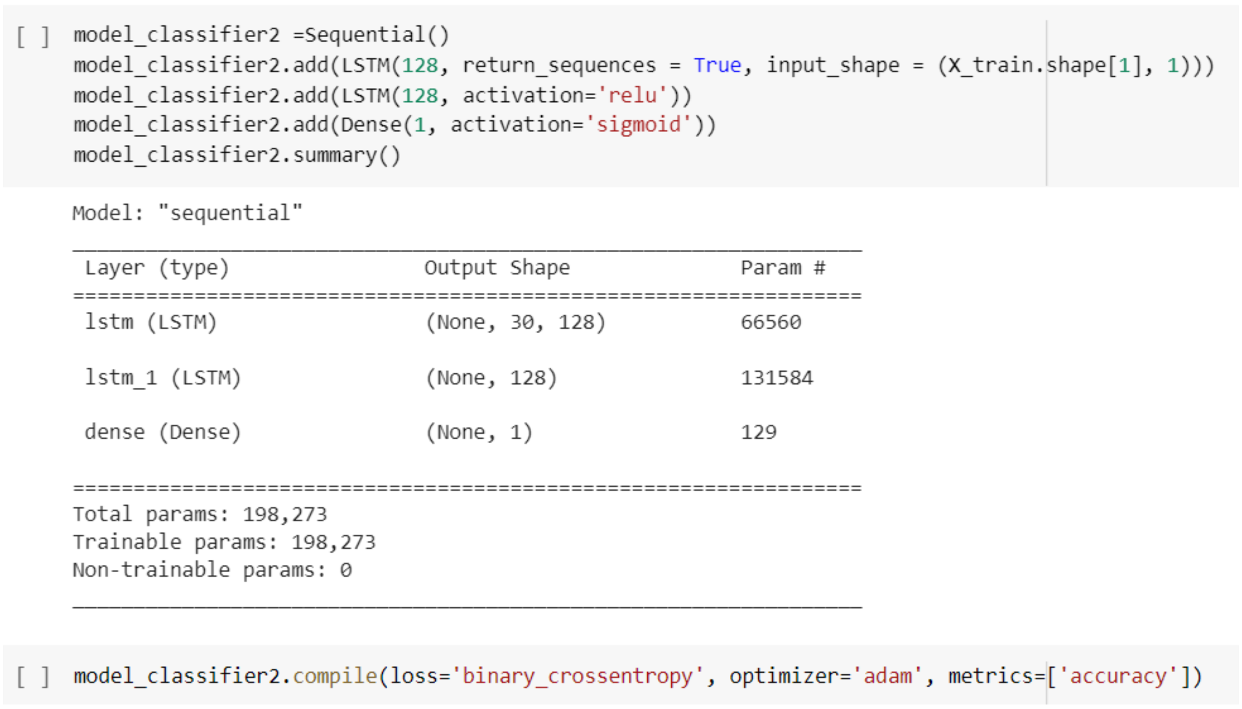


```
def word_tokenize(text):
    return nltk.word_tokenize(text)
```

Screenshot 3: Tokenizing Function

Next step is to tokenize the words. So a function is defined to tokenize the text with the help of the NLTK library.

## 3. Classification Model

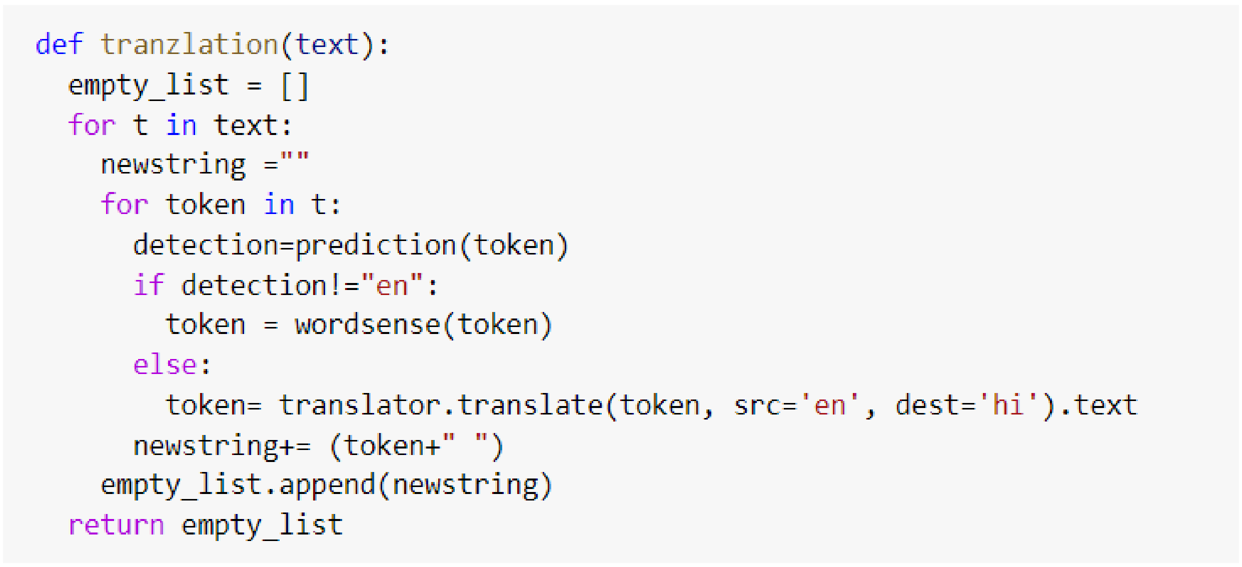


```
[ ] model_classifier2 =Sequential()
    model_classifier2.add(LSTM(128, return_sequences = True, input_shape = (X_train.shape[1], 1)))
    model_classifier2.add(LSTM(128, activation='relu'))
    model_classifier2.add(Dense(1, activation='sigmoid'))
    model_classifier2.summary()

    Model: "sequential"
    _________________________________________________________________
     Layer (type)                Output Shape              Param #
    =================================================================
     lstm (LSTM)                 (None, 30, 128)           66560

     lstm_1 (LSTM)               (None, 128)               131584

     dense (Dense)               (None, 1)                 129

    =================================================================
    Total params: 198,273
    Trainable params: 198,273
    Non-trainable params: 0
    _________________________________________________________________

[ ] model_classifier2.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Screenshot 4: Classification model

A three layer LSTM model was used to classify tokens as hindi or english. This model made use of relu activation functions in the first 2 layers and sigmoid activation function in the last layer.The loss function used was binary cross entropy and the optimizer was adam.

## 4. Classifying and performing Transilteration /Translation Operation

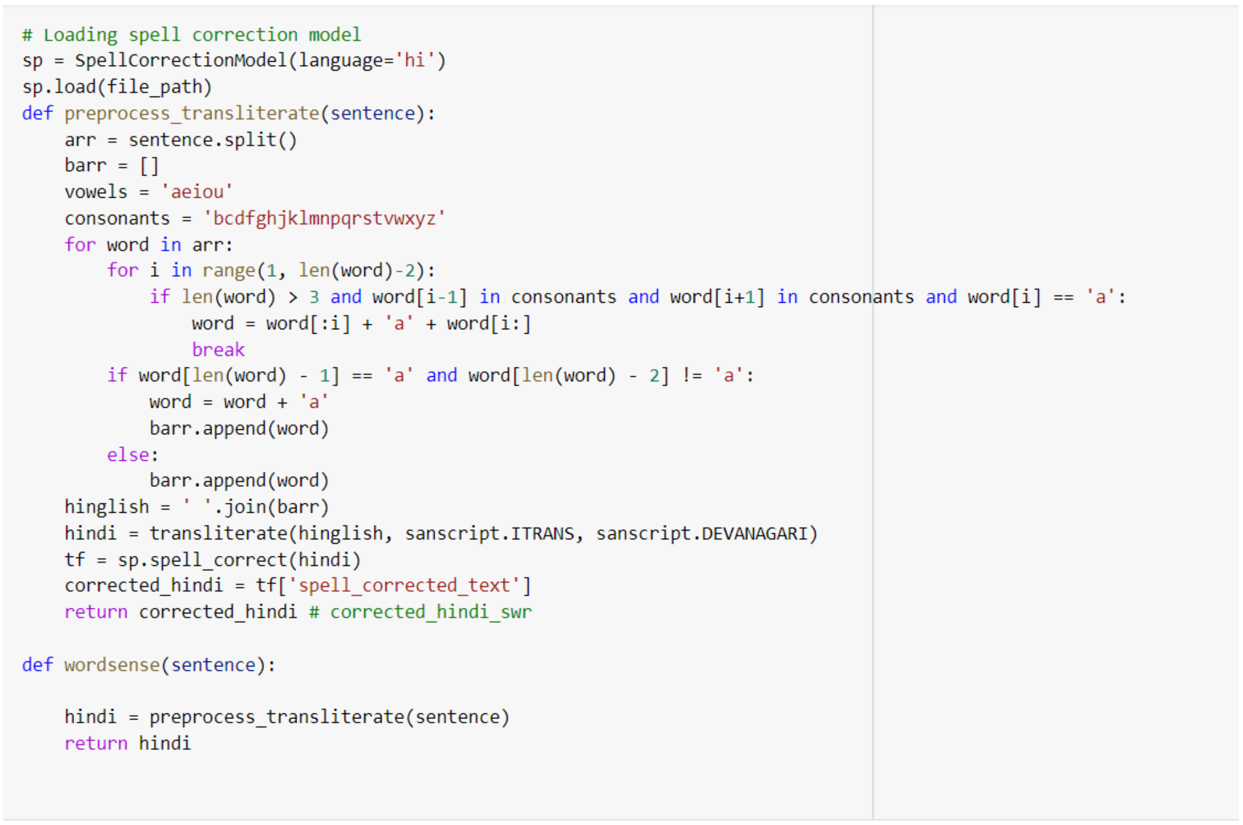


```
def tranzlation(text):
  empty_list = []
  for t in text:
    newstring =""
    for token in t:
      detection=prediction(token)
      if detection!="en":
        token = wordsense(token)
      else:
        token= translator.translate(token, src='en', dest='hi').text
      newstring+= (token+" ")
    empty_list.append(newstring)
  return empty_list
```

Screenshot 5: Translation function

The translation function first receives the sentence as an input and then tokenizes it, the tokens are then classified as either "hi" or "en" using the LSTM model as described above. The tokens that are classified as "hi" are sent to the wordsense function where they are transliterated into Devnagri Hindi,and the tokens classified as. "en" are translated into Devanagari Hindi using the Googletrans package.Finally both these tokens are appended to output a string.

## 5. Transliteration

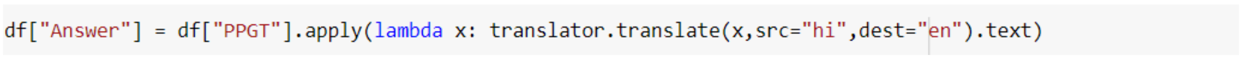


```python
# Loading spell correction model
sp = SpellCorrectionModel(language='hi')
sp.load(file_path)
def preprocess_transliterate(sentence):
    arr = sentence.split()
    barr = []
    vowels = 'aeiou'
    consonants = 'bcdfghjklmnpqrstvwxyz'
    for word in arr:
        for i in range(1, len(word)-2):
            if len(word) > 3 and word[i-1] in consonants and word[i+1] in consonants and word[i] == 'a':
                word = word[:i] + 'a' + word[i:]
                break
        if word[len(word) - 1] == 'a' and word[len(word) - 2] != 'a':
            word = word + 'a'
            barr.append(word)
        else:
            barr.append(word)
    hinglish = ' '.join(barr)
    hindi = transliterate(hinglish, sanscript.ITRANS, sanscript.DEVANAGARI)
    tf = sp.spell_correct(hindi)
    corrected_hindi = tf['spell_corrected_text']
    return corrected_hindi # corrected_hindi_swr

def wordsense(sentence):

    hindi = preprocess_transliterate(sentence)
    return hindi
```

Screenshot 6: Transliteration function

Tokens classified as "hi" are sent into this function and transliterated using the Indic Transliteration package after making minor changes to the hinglish spellings.The transliterated token is then passed through the Hindi spello model,that corrects its spelling in Devnagri Hindi.

## 6. Translation to English

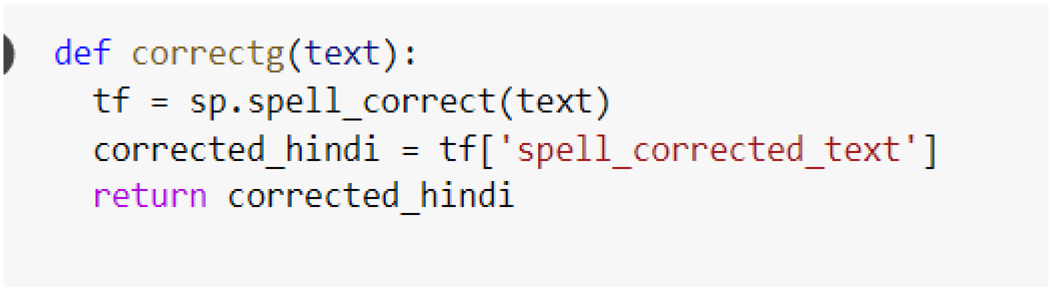


```python
df["Answer"] = df["PPGT"].apply(lambda x: translator.translate(x,src="hi",dest="en").text)
```

Screenshot 7: Translation

The devnagri Hindi sentence obtained is passed through the Google trans package and converted to English.

## 7. Spelling Correction

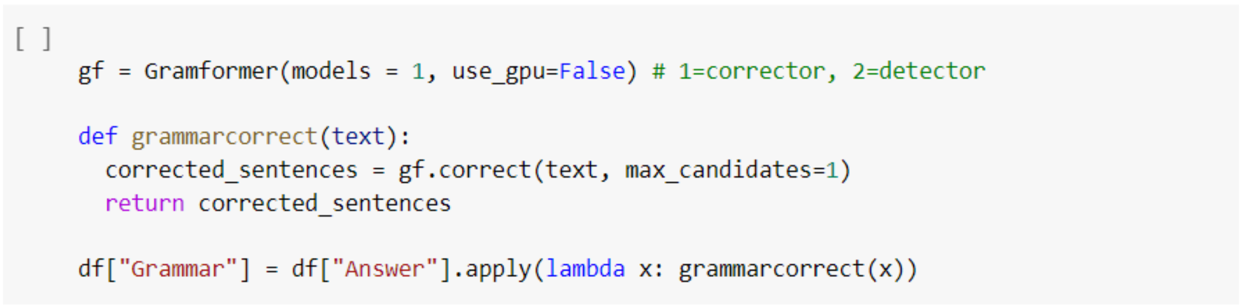


```python
def correctg(text):
  tf = sp.spell_correct(text)
  corrected_hindi = tf['spell_corrected_text']
  return corrected_hindi
```

Screenshot 8: Correcting spelling of translated text

In order to ensure that the transliterated text follows the spelling conventions of Devanagari Hindi, the text is corrected using the Spello Model.

## 8. Grammar Correction

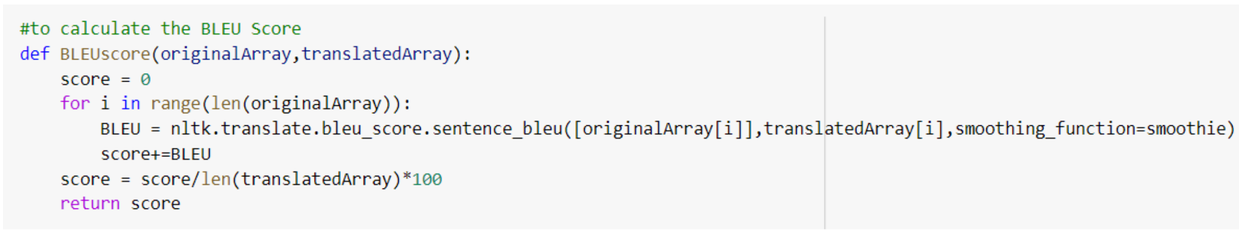


```
[ ]
    gf = Gramformer(models = 1, use_gpu=False) # 1=corrector, 2=detector

    def grammarcorrect(text):
      corrected_sentences = gf.correct(text, max_candidates=1)
      return corrected_sentences

    df["Grammar"] = df["Answer"].apply(lambda x: grammarcorrect(x))
```

Screenshot 9: Correcting grammar of translated text

The text translated into English was found to have several Grammatical Errors, Therefore the grammar in these sentences was checked and corrected using the Gramformer Model.

## 9. Evaluation Using BleU score



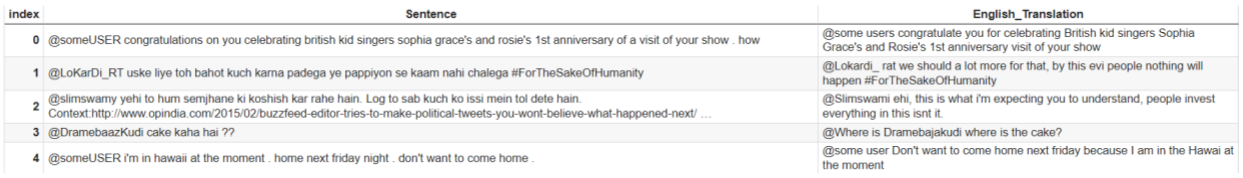
```
#to calculate the BLEU Score
def BLEUscore(originalArray,translatedArray):
    score = 0
    for i in range(len(originalArray)):
        BLEU = nltk.translate.bleu_score.sentence_bleu([originalArray[i]],translatedArray[i],smoothing_function=smoothie)
        score+=BLEU
    score = score/len(translatedArray)*100
    return score
```

Screenshot 10: Evaluating using BleU score

In this function we calculate the BleU scores of each sentence in the corpus, and the sum of these scores is converted into a cumulative score. That gives us our final result.

## V. Screenshots

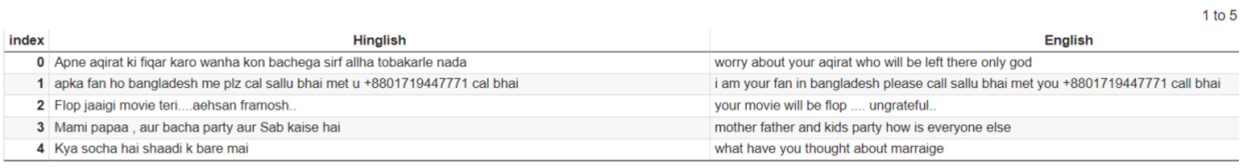
We used two code-mixing datasets released by Dhar et al. (2018) [9] and Srivastava and Singh (2020) [1] for this work. These datasets were chosen for their variety of sources (all major social networking platforms), which reduces bias in the dataset. The dataset produced by Srivastava and Singh (2020) [1] contains 13,760 rows of hinglish social media comments and tweets transformed to English counterparts.This dataset consists of 103,887 hinglish tokens and 96,439 english and other tokens.Following are the first five entries of the dataset that comprise the hinglish sentence and its English translation:



| index | Sentence | English_Translation |
|---|---|---|
| 0 | @someUSER congratulations on you celebrating british kid singers sophia grace's and rosie's 1st anniversary of a visit of your show . how | @some users congratulate you for celebrating British kid singers Sophia Grace's and Rosie's 1st anniversary visit of your show |
| 1 | @LoKarDi_RT uske liye toh bahot kuch karna padega ye pappiyon se kaam nahi chalega #ForTheSakeOfHumanity | @Lokardi_ rat we should a lot more for that, by this evi people nothing will happen #ForTheSakeOfHumanity |
| 2 | @slimswamy yehi to hum semjhane ki koshish kar rahe hain. Log to sab kuch ko issi mein tol dete hain. Context:http://www.opindia.com/2015/02/buzzfeed-editor-tries-to-make-political-tweets-you-wont-believe-what-happened-next/ ... | @Slimswami ehi, this is what i'm expecting you to understand, people invest everything in this isnt it. |
| 3 | @DramebaazKudi cake kaha hai ?? | @Where is Dramebajakudi where is the cake? |
| 4 | @someUSER i'm in hawaii at the moment . home next friday night . don't want to come home . | @some user Don't want to come home next friday because I am in the Hawai at the moment |

Screenshot 11: Dataset of Shrivastava and Singh

Dhar et al. (2018) [9] developed a dataset for machine translation of code-mixed data that included 6,096 English-Hindi code-mixed and English monolingual gold standard parallel sentences. This dataset was used for machine translation of code-mixed data.There are a total of 37,673 Hinglish tokens and 26,276 other tokens ,including english tokens.contained within this dataset.The hinglish statement and its translation in English are presented in the following format across the first five rows of the dataset

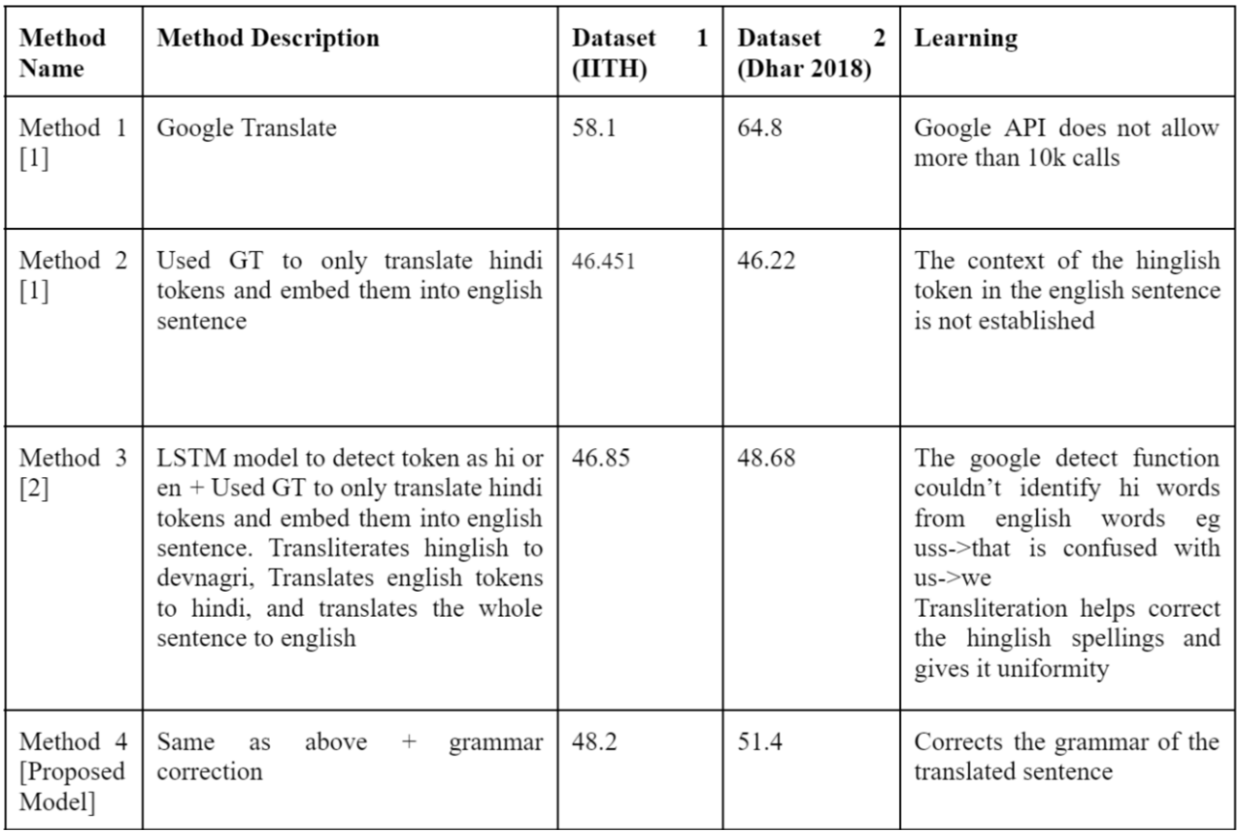


1 to 5

| index | Hinglish | English |
|---|---|---|
| 0 | Apne aqirat ki fiqar karo wanha kon bachega sirf allha tobakarle nada | worry about your aqirat who will be left there only god |
| 1 | apka fan ho bangladesh me plz cal sallu bhai met u +8801719447771 cal bhai | i am your fan in bangladesh please call sallu bhai met you +8801719447771 call bhai |
| 2 | Flop jaaigi movie teri....aehsan framosh.. | your movie will be flop .... ungrateful.. |
| 3 | Mami papaa , aur bacha party aur Sab kaise hai | mother father and kids party how is everyone else |
| 4 | Kya socha hai shaadi k bare mai | what have you thought about marraige |

Screenshot 12: Dhar et al. Dataset

## VI. Results & Discussion



| Method Name | Method Description | Dataset 1 (IITH) | Dataset 2 (Dhar 2018) | Learning |
|---|---|---|---|---|
| Method 1 [1] | Google Translate | 58.1 | 64.8 | Google API does not allow more than 10k calls |
| Method 2 [1] | Used GT to only translate hindi tokens and embed them into english sentence | 46.451 | 46.22 | The context of the hinglish token in the english sentence is not established |
| Method 3 [2] | LSTM model to detect token as hi or en + Used GT to only translate hindi tokens and embed them into english sentence. Transliterates hinglish to devnagri, Translates english tokens to hindi, and translates the whole sentence to english | 46.85 | 48.68 | The google detect function couldn't identify hi words from english words eg uss->that is confused with us->we<br>Transliteration helps correct the hinglish spellings and gives it uniformity |
| Method 4 [Proposed Model] | Same as above + grammar correction | 48.2 | 51.4 | Corrects the grammar of the translated sentence |

Table 2: Comparison study of all the methods

### 1. Challenges

- **Hindi is Context Dependent :**

  A large number of hinglish terms have several meanings that can be determined from the sentence context alone [1]. An example of this could be the term "chalega" which in some sentences means "will work" and in some sentences means "walk".

- **Lack of Standardization:**

  There are no standard spellings in Hinglish; most users rely on the phonetics of the word to determine its romanized spelling [1], thus resulting in a variety of words with the same meaning but different spellings. Eg "Nahi", "Nai", "Nhi" all mean "No" in english.Another example could be "main, mai, mein" all meaning "me" in english.

- **Unstructured Data:**

  Hinglish is most frequently encountered on informal platforms. Since informal writing rarely adheres to punctuation, correct spelling, and correct grammar, this adds an additional layer of complication to the translations. In addition, the use of abbreviated words might lead to misunderstanding on whether a word is hindi or English.

- **Similar words in Hindi and English:**

  Several words used in Hinglish are also found in English [2], in such cases it creates an obstacle for the model to determine the source language and thus choose between transliterations and translations. Eg the term "main" can be used in both hinglish where it means "me" and english where it means "principal".

- **Lack of Processing Tools, Packages and Techniques:**

There is a severe lack of research and large-scale technologies that can be used to efficiently produce POS tags, Named Entity Recognition, or even Word Embeddings because much of the research on Natural Language Processing is done mostly in English [10], with some of it in monolingual languages. This makes it difficult to design large-scale algorithms that can analyse, classify, or determine the sentiment of Hinglish text, among other things.

- **Availability of Data:**

  Since Hinglish is typically spoken in casual conversational contexts, the vast majority of relevant data is not publicly available and hence cannot be acquired [10]; yet, its availability would greatly facilitate the resolution of the issues discussed.

## VII. Conclusion

This paper presents an implementation of code-mixed translation strategies for Hinglish to English translation. Our learnings from the implementations have led us to develop a framework that represents an improvement over previous methods, although it may not yet surpass Google's Translation API. Furthermore, this research highlights several obstacles that must be overcome to develop truly effective code-mixed machine translation systems. The insights gained from this study can be extended to other code-mixed text, expanding the potential applications of our work.

In the future, our research may involve training machine translation models such as mBART and mT5 on the datasets we have created. However, one major challenge remains the scarcity of code-mixed Hinglish data[3]. To address this, future research could focus on generating larger parallel corpuses[6]. Additionally, there is a need for pre-trained code-mixed translation models, as training models like mBART from scratch is computationally demanding.

Therefore, this study improves the understanding of code-mixed machine translation and proposes research topics that will facilitate the creation of more robust translation systems.

## References

1. V. Srivastava and M. Singh, “Papers with Code - PHINC: A Parallel Hinglish Social Media Code-Mixed Corpus for Machine Translation,” PHINC: A Parallel Hinglish social media Code-Mixed Corpus for Machine Translation | Papers with Code. [Online]. Available: https://paperswithcode.com/paper/phinc-a-parallel-hinglish-social-media-code. [Accessed: Oct. 31, 2022]
2. I. Jadhav, A. Kanade, V. Waghmare, S. S. Chandok, and A. Jarali, “Code-Mixed Hinglish to English Language Translation Framework,” Code-Mixed Hinglish to English Language Translation Framework. [Online]. Available: https://ieeexplore.ieee.org/document/9760834. [Accessed: Oct. 31, 2022]
3. D. Gautam, K. Gupta, and M. Shrivastava, “Translate and Classify: Improving Sequence Level Classification for English-Hindi Code-Mixed Data | Semantic Scholar,” | Semantic Scholar, Jan. 01, 2022. [Online]. Available: https://www.semanticscholar.org/paper/Translate-and-Classify%3A-Improving-Sequence-Level-Gautam-Gupta/717337fd461d3f32894d2a1e87a2b3b3bdc22cb7. [Accessed: Oct. 31, 2022]
4. Approaches for Improving Hindi to English Machine Translation System
5. S. H. Attri, T. V. Prasad, and G. Ramakrishna, “HiPHET: A Hybrid Approach to Translate Code Mixed Language (Hinglish) to Pure Languages (Hindi and English) | Computer Science,” HiPHET: A Hybrid Approach to Translate Code Mixed Language (Hinglish) to Pure Languages (Hindi and English) | Computer Science, Sep. 27, 2020. [Online]. Available: https://journals.agh.edu.pl/csci/article/view/3624. [Accessed: Oct. 31, 2022]
6. Towards Translating Mixed-Code Comments from social media
7. Hinglish to English Machine Translation using Multilingual Transformers
8. S. N. Bhattu and V. Ravi, “Language Identification in Mixed Script Social Media Text.” Fire workshops, 2015, pp. 37-39.
9. M. Dhar, V. Kumar, M. Shrivastava, “Enabling code-mixed translation: Parallel corpus creation and mt augmentation approach”, Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing, 2018, pp131-140.
10. Srivastava, V., & Singh, M. (2021). Challenges and considerations with code-mixed nlp for multilingual societies. arXiv preprint arXiv:2106.07823.