# Project Outline

## Team: Amigos ( 2020101103, 2020102061, 2020101077 )
## CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences

---

**What is Code Mixed Languages?**

Code-mixing is the mixing of two or more languages or language varieties in speech. In other words, it is the embedding of linguistic units such as phrases, words, and morphemes of one language into an utterance of another language.

- In a Multilingual society, the increase in the population and increased usage of social media resulted in various forms of informal writing. One of them is code-mixed languages.
- From my personal experience, I grew up speaking Telugu and English and have a fair command of Hindi. Even though I usually speak Telugu at home, I always notice other languages tend to find their way into my conversations.

**Purpose/Aim:**

Generating a machine translation system for English to Hinglish in supervised learning. In supervised learning, the algorithm "learns" from the training dataset by iteratively making predictions on the data and adjusting for the correct answer. (Here we are making use of supervised learning because we are using labeled data so that we can get the entire context at the encoder.)

- The translation systems could be used to augment datasets for various Hinglish tasks by translating datasets from English to Hinglish.
- The above-described model has to do the below translation from English to Hinglish :

Let's take an example say,

Example 1:

Hinglish sentence:     *Main kal* movie *dekhne jaa rahi thi* and *raasthe me* I met Ram.

English sentence:     I was going for a movie yesterday and on the way, I met Ram.

Example 2:

Hinglish sentence:     *Bahut* strange choice *thi ye.*

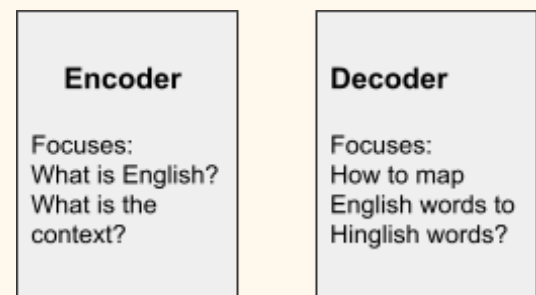English sentence:     This was a very strange choice.

# Problem:

The informal nature of this language (code mixed) makes it difficult to collect code mixed data which is a major contributing factor.

- In addition to the use of language, code-switching also involves switching between dialects, styles of speech, gestures, body language, and vocal registers. So it's difficult to collect data.

Another challenge with Hinglish, in particular, is that there is no standard system of transliteration for Hindi words.

# Solution:

Before heading towards the solution let's get a complete idea of BART. It is similar to that of transformers as it includes encoders and decoders. BERT (Bidirectional Encoder Representations from transformers), Random tokens are replaced with masks, and the document is encoded bidirectionally missing tokens are predicted independently. So BERT cannot easily be used for generations. In mBART, we make use of two models, the Masked language model and the Next Sentence Prediction.

**Encoder**

Focuses:
What is English?
What is the context?

**Decoder**

Focuses:
How to map
English words to
Hinglish words?

### Implementation:

- Pre-train the BERT to understand the language.
- Fine-tune BERT to learn a specific task. (Here in our model, that specific task is to translate to Hinglish).
- So, we first fine-tune mBART by first transliterating the Hindi words in target sentences from Roman script to Devanagari scripts to utilize its pre-training. (we make use of Roman scripts because it helps the encoder with more contextual analysis). Then we further translate the English input to Hindi using pre-existing models and show improvements in the translation using parallel sentences as input to the mBART model.

**How can we input parallel sentences to the mBART model?**

Considering the below paper as a reference, we can input parallel sentences to the mBART model.

https://aclanthology.org/2021.eacl-demos.24/

- Code-mixing is common in multilingual communities around the world, and processing it is challenging due to the lack of labeled and unlabeled data. We describe a tool that can automatically generate code-mixed data given parallel data in two languages.
- The toolkit can be used to generate unlabeled text data for pre-trained models, as well as visualize linguistic theories of code-mixing.

**Model:**

We use mBART, the below are the two strategies we compare for proper translation: one focuses on translating the English sentences to Hinglish sentences, and the other uses the two English and Hindi Sentences for translating to Hinglish.

    (a) mBART-en: We fine-tune (as discussed above) mBART on the train set, where English sentences are given as input to an encoder and whereas Hinglish sentences are the decoder output.

    (b) mBART-hien: We fine-tune mBART on the train set, here parallel sentence input is given as input to the encoder i.e., both English sentences along with parallel Hindi translation and we get the Hinglish sentences as the output.

- As the input we have is only the English sentences, but for mBERT-hien we need Hindi sentences, so we use NMT(Neural Machine Translation) which is an end-to-end learning approach for automated translation. For the input, we concatenate the Hindi sentence and English sentences separated by '##'.
Reference for NMT: https://arxiv.org/abs/1609.08144

Code related:

```
if MODEL == 'mBARTen':
    src_data.append(' '.join(obj['English']) + '\n')
elif MODEL == 'mBARThien':
    src_data.append(' '.join(obj['Hindi']) + ' ## ' + '
'.join(obj['English']) + '\n')
```

We will use Adam Optimizer with suitable dropout and learning rate.

## Dataset:

We are using the dataset from the following link:

https://ritual.uh.edu/lince/datasets

This dataset contains English-Hinglish.

Ex: Not very popular it seems; lagta hai bahut popular nahi hai

- In mBART-hien, English sentences and translated Hindi sentences. So, as covered before we use NMT for that.

**Statistics of the dataset**

|  | Train (From the train.txt) | Valid (From the dev.txt) |
|---|---|---|
| # of sentences | 8060 | 942 |
| # of tokens in source sentences | 98080 | 12275 |
| # of tokens in target sentences | 101752 | 12611 |

| # of Hindi tokens in target sentences | 68054 | 8310 |
|---|---|---|
| # of English tokens in target sentences | 21502 | 2767 |
| # of other tokens in target sentences | 12196 | 1534 |

# Evaluation Metrics:

How to evaluate the performance of the Machine Translation system? The only solution is human evaluation. But the problem is that this is a time taking process. The optimized way is using BLEU.

**BLEU:** Bi-Lingual Evaluation Understudy

It compares a machine translation with existing human-generated translation (reference translations).

- Calculation: BLEU is calculated based on precision. The ratio of the covered candidate words out of the Total number of candidate words. Then we sum that for the entire sentence i.e., BLEU* = (D(w1) + D(w2) + D(w3) + ...... +D(wn))/n
- In coding, we are using SacreBLEU library to calculate BLEU score. https://github.com/mjpost/sacrebleu

## BLEU Normalized:

Similar to the calculating process of the BLEU as above but the only difference is calculating the scores on texts where Hindi words are in Devanagari and English words in Roman. But whereas in the above BLEU we calculate the scores on the texts where the Hindi words are transliterated to Roman.

- We are going to use these two evaluation metrics for comparing our systems.

# Resources:

1. **To input parallel sentences to the mBART model:**
   https://aclanthology.org/2021.eacl-demos.24/
2. **NMT(Neural Machine Translation) used in mBART-hien:**
   https://arxiv.org/abs/1609.08144
3. **Library used for BLEU score calculation:**
   https://github.com/mjpost/sacrebleu

# TIMELINE

**Oct 6 - Oct 13**
Get complete clarity of the entire project and flow of the code . And will start working on dataset part.

**Oct 14 - Oct 21**
Continue working on the dataset part and working on modelling mBART-en

**Oct 22 - Oct 30**
Making sure on proper working of the model. Then focusing on mBART-hien modeling. We make use of NMT for this model.

**Oct 30 - Nov 10**
Working on the Evaluation Metrics using BLEU and BLEU Normalization.

**Nov 11 - Nov 16**
Re-check the entire working and also the BLEU scores