Group ID: G-02
Team Members: Adi Kamaraj, Aditya Ganesh Kumar, Lee Rong Jieh, Li Jin Tao Joey, Niharika Shrivastava
Project Topic: Grammatical Error Correction (GEC)
Task Description: GEC deals with correcting writing errors in a written text in English. These errors include spelling errors, punctuation errors, grammar errors, word choice errors, etc.
System Input/Output: Grammatically erroneous sentences / Grammatically corrected sentences
Prior Related Work Outline:

- **Performance of state-of-the-art (SOTA) GEC systems:** Grammatical Error Correction | NLP-progress
- **Baseline System:** Satoru Katsumata and Mamoru Komachi Stronger Baselines for Grammatical Error Correction Using a Pretrained Encoder-Decoder Model.

Training and Development Data

- FCE v2.1: Link, Lang-8 Corpus of Learner English: Link, NUCLE (NUS Corpus of Learner English) and W&I+LOCNESS v2.1: Link

The training data files are in the M2 format. Essentially, each .m2 file consists of blocks separated by a blank line between two blocks. Each block consists of one tokenized sentence, followed by annotations of the errors found in the sentence. Each annotated error includes the start token position, end token position, error type, and the replacement string.
Test Data

- CoNLL-2014, BEA, JFLEG and BEA-2019 shared task.

Approach:

- Use a pre-trained model (BERT, BART, GPT3, Google) as a base. Add character-level noise to the training set and provide this as input to the LM model. Integrate a LM based on bigrams/trigrams. Do 1 iteration of prediction. Then fine-tune hyperparameters. Try ensemble of different pre-trained models with the LM-based integration. Fine-tune the ensemble

Any external (supporting) code to be used?
Source code for GPT-3 - (https://github.com/openai/gpt-3), Rest to be decided
Breakdown of Work:

Model: Aditya Ganesh Kumar, Adi Kamaraj
Report: All
Transformer: Rong Jieh, Niharika Shrivastava
LM/RB: Rong Jieh, Joey Li
Test: Joey Li, Niharika Shrivastava

Schedule/Timeline:

- Week 8: Literature survey
    - Transformers (Encoder-Decoder models), Different pretrained models, LM-based approaches based on n-grams, Ensemble Learning
- Week 9: Create and verify baseline system
    - Define model architecture
- Week 10 - 11: Train - Validate
    - Test
- Week 12: Report and conclusion