Milestone2

1. **Research question**: Evaluate how different tokenization strategies impact the performance and efficiency of large language models (LLMs) in multilingual tasks.
    a. Quantify performance metrics (e.g., BLEU, F1-score).
    b. Measure efficiency trade-offs (model size, computational cost, inference speed).
2. Related Work:
    a. Analyze Sennrich et al. (2016) for insights into handling rare words with subword units.
    b. Study Kudo (2018) to understand subword regularization and its effect on neural machine translation models.
3. Method:
    a. Tokenization Strategies
        i. Byte Pair Encoding (BPE).
        ii. Unigram Language Model.
        iii. Subword Sampling.
    b. Multilingual Datasets
        i. Dataset Selection Criteria:
            1. Language Diversity: Include high-resource (e.g., English, Chinese) and low-resource (e.g., Swahili, Urdu) languages.
        ii. Tasks:
            1. Translation: Use datasets like WMT.
            2. Sentiment Analysis: Use datasets like Multilingual Amazon Reviews or XNLI.
        iii. Preprocess datasets to standardize input for each tokenization strategy.
    c. LLM Training/Fine-tuning
        i. Model Selection: Start with a pre-trained LLM (e.g., mBERT, XLM-R, or BLOOM) to reduce computational cost and fine-tune models for each tokenization strategy on the selected datasets.
        ii. Training Protocol:
            1. Keep hyperparameters constant across experiments.
            2. Train/fine-tune models to convergence or for a fixed number of epochs.
            3. Log metrics during training (e.g., loss, validation accuracy).
    d. Evaluation Metrics
        i. Translation tasks: BLEU Score: Measures translation quality.
        ii. Sentiment analysis: F1-Score: Measures classification performance.
4. Preliminary result:
    a. Right now, we are finished reading the related works and are fully understand how those three tokenization methods work.
    b. We are moving forward to datasets and model selection phrase.
5. Preliminary discussion + question for the Tas
    a. For CPU-only training, because I only have access to my Macbook, is this feasible? We should choose relatively small model and small dataset, right? Do

you have any recommendation for small pre-trained model? Or small datasets for translation and sentiment test?

b. For the detailed step-by-step method, we are using right now, is there any place where we can modify? Is this plan sounds good?