# CS410 Project Progress Report

Felipe Assumpção, Rafael Piacsek

November 2023

## 1 Progress Made

So far, we have finished all the tasks related to high-level design choices, data preprocessing, and all setup logic/functionality required to properly design, train, and improve a functional model for the project.

- **Design Choices:** We had a long discussion regarding some different designs and implementation ideas with the data we had and what we wanted our final program to generate. We first decided that we would only need one model that would classify different segments of the input text (essentially combining the two segmentation and classification models we originally had in mind). Additionally, we decided that the best way to label the different segments (in both training data and for the model output) would be with NER labels, where we label every token as either being the beginning of a new discourse type (a segment with a different classification), inside some discourse type, or unannotated.
- **Data Preprocessing:** A large part of our time was spent parsing and manipulating the original data available to us in order to create a new dataset that only contained information the model would need and in an appropriate format. The original dataset had a separate entry for each discourse/segment, which meant we had to find all entries that belonged to a single text and properly map the position of every token in each discourse to its position in the overall text to create a final list of NER labels that we would use. The final dataset is simply a data frame of all texts, their IDs, and a list of corresponding labels for each token in the text.
- **Backend Setup for Model:** Finally, we finished defining the PyTorch dataset functions we will need and use, as well as creating Dataloaders for training and validation. Crucially, these allow us to train our model using different configurations, parameters, and tokenizers from pre-trained models in order to boost model performance. Additionally, it will be very easy to experiment with different values and configurations in the future when we want to optimize our model.

## 2 Remaining Tasks

- **Model Functionality and Metrics:** We still need to design and implement several functions for our model, such as a train function and what the model does during each training step, an inference function, and any functions for getting metrics to properly measure our model's accuracy and performance. We also need to design the actual model architecture, which we can do from scratch or finetune a pretrained model for better results.
- **Model Optimization:** After implementing everything related to the model and getting baseline results, we will spend some time experimenting with different models and configurations as well as perform hyperparameter tuning until we reach what we believe is an ideal accuracy and level of performance for our model.
- **Command Line Interface:** Finally, we will create a command line interface for users to easily input their own texts and generate a list of predicted labels of different discourses/segments. Depending on the time we have left, we can expand this into a web app to give people a better experience when getting predictions from the model.

## 3 Challenges

The biggest challenge we faced thus far was that we spent way more time than we estimated on choosing what we thought was the best approach for our problem between several possible design choices we could take. Once we fetched and parsed the data, several different possibilities for exactly how the model could use it came to mind that we hadn't even thought about before, so we spent a lot of time researching and discussing what the best course of action was to ensure these high-level choices set us up for a successful project in the end.