This semester, Ismini and I continued our work on controversy-detection within news articles. Much progress was made in all areas, but most notably were efforts in building a dataset, improving the scoring algorithm, and implementing more precise retrieval of social content. In the beginning of the semester, the demo [1] was also improved.

Our work in building a dataset for training was the most apparent progress. The unannotated dataset consists of **X** unique news articles including their metadata and comments, as well as relevant tweets. Retrieval of some tweets and all news articles is keyword-based. Thus, a single term is used to query and return a list of up to 10 news articles and 500 tweets. These tweets and the URL of each article as well as attributes such as date, author, and the article's abstract are retained. We then scrape the article's webpage to obtained the full-text of the article. Finally, a second twitter query is performed using the article's title without stopwords. All of these data are saved within a schema to a persistent database. Non-tweet content is obtained from The New York Times, Reuters, and the Associated Press. Article webpage comments contain many metadata variables, all of which we save. In all twitter queries, we also save metadata related to the tweet itself (retweets, favorites, etc.) and that related to the author (location, followers, etc.). All of this content is stored in a persistent database and is optimized for training.

The largest obstacle in curating the unannotated dataset was speed. Given the considerable size of the data (and single-core nature of the server!), building an efficient scraper was key. However, time to finish this aspect of the project and return to improving the scoring function was equally important. Speed in the former case was improved by not running various analyses on the social content retrieved. We do as little post-processing as possible: no scoring occurs, no linguistic or sentiment features are noted, and organizing into a schema-based database is avoided. Another consideration was not using a ranking function such as Okapi BM25 to filter tweets only by their relevance to the articles.

[1] https://controversy.2pitau.org