Personal Report

Capstone Project 2019 Spring: Stance Detection and Stance Classification
Author: Xiaochi Li

My contribution in the project

1. **Data Loading**
   Design and developed *corpus_loader.py* for loading speech with labels, *xml_parser.py* for parsing XML files.
2. **Relabeling data with regular expression**
   Observed 200 speech and developed *regex_stance_detection.py* for relabeling unlabeled speech. Validate the regex model on labeled data to find the best cutoff point.
3. **Stance Classification**
   Design and developed *Pipeline_V1.py*, a one-stop pipeline for preprocessing and training model with minimum code to call.
   Developed *Experiment_Framework.ipynb*, a framework that only needs minimum change to test different model on 12 combinations of preprocessing, vectorization and balancing data.
   Developed mean_embedding_vectorizer.py, a new vectorizer class to combine mean embedding of word2vec and bag of words embedding. And tested its performance.
4. **Exploratory Data Analysis and Experiment**
   Performed exploratory data analysis and tested several preprocessing methods and model in Model_V1_Stance_Detection.ipynb and Model_V1_Stance_Classification.ipynb. These two files are not well organized, only used for reference.
5. **Visualization**
   Drew graphs for the presentation, developed *plot_function.py* to draw bar chart of experiment result.
6. **Parallel Computing**
   Developed parallel_computing.py and used python's multiprocessing library to speedup data preprocessing process.

My learning

1. Project Management
   Thanks to our mentor Dr. Daniel Argyle from FiscalNote, we can always have his support and guidance during the project.
   We took project management and agile methodology into practice. In the first meeting, we split our project into three parts (loading data, stance classification and stance detection). Then we applied the Agile methodology in the daily work. We plan the short-term plan for the day or the week during daily morning meeting and every day is an agile sprint of plan-

development-test.

We found this “rapid iterative failure” method is quite useful to testing new things. Since there is no best model for all problem, we can not fix on one method for too long. The Agile methodology makes us know which methods are useless and focus on methods that has good potential.

2. Communication

   The daily meeting with Daniel is a good chance for me to practice how to communicate my work in a concise and efficient way. Also, I learnt to communicate my plan with my teammate.

3. Program Design

   Since we are doing a group project, the efficiency can be improved when the program can be easily reused by the other person. My principle of design these common programs like corpus_loader.py and Pipeline_V1.py is to make the person who will use the code need to write as few lines as possible.

   For example, the usage of corpus loader could be only one line like

   ```
   tagged_df = corpus_loader(parser='bs')
   ```

   And try a combination of methods with pipeline_v1 is like:

   ```
   Pipeline(X, y, vectorizer=count, model=model, sampler=None)
   ```

4. Think / Observe / Plan before do

   There are two places in the project where I started coding before thought through. One place is when designing the XML parser, I used python’s xml library before searching other ways. And it took me one day to build an XML parser for scratch. Then I found beautifulsoup4 library can do the same job, and it only needs 30 minutes to develop based on beautiful soup.

   Another place is the stance detection part, I spent 2 weeks on tuning the model before I observed the data and found it needs relabeling.

   And the lesson I learnt is that I should always think/plan/observe before do something. The slower way may be faster in the end.

5. Feature engineering sometimes is more important than modeling

   In the project, one of the finding is that the model can improve greatly when we do the right preprocess. It’s an important lesson.

6. Needs to consider time to process

   The problem of data size did not occur when we only two congressional sessions. However, when we face 20 sessions, that will be a problem. For example, we need 12 hours to do the lemmatization on 5 sessions of record. The current size of data is just several hundreds of MB. I think it’s necessary to learn some big data tools to get ready for a larger data size in the future.