CS51 Project: Part-of-Speech Tagging and Highlighter Technical Specification

Team

- Billy Janitsch, janitsch@college.harvard.edu
- Jenny Liu, jennyliu@college.harvard.edu
- Yuechen Zhao, yuechenzhao@college.harvard.edu
- Shijie (Joy) Zheng, shijiezheng@college.harvard.edu

Signatures/Interfaces

(See javadoc files in GitHub repository)

Modules/Actual Code/Progress Report

We have organized all of the fundamental functions of our modularized program into a file directory in which all code will reside. This has largely reduced the project to a matter of implementing discrete functions.

We have thoroughly discussed many of the intricacies of our program (particularly with regards to the POS tagger and the HMM) including corner cases, which should to a large extent eliminate difficulties later in the coding process. This also involved explicitly writing out the form of any save files our program generated as well as the expected form of any input files. We have also agreed on how specifically the GUI will look and function to the user.

After trying out several data structures (Arrays, LinkedLists, Vectors, ArrayLists, HashMaps), we realized both that, in order to avoid excessive copying while maintaining constant-time lookup in tables/lists during training, we would need to load the set of parts of speech going in. Additionally, while we were fleshing out more of the methods and the control flow, we realized that the Dictionary class was being used primarily for training, rather than for actual lookups, since the probabilities in the Viterbi algorithm would track which parts of speech a word could correspond to, anyways. As a result, we merged the former Dictionary and Viterbi classes into a single Viterbi class and sketched out the main actions and function calls inside. (can be seen in the file Viterbi.java)

We also significantly fleshed out the POS class, adding a name field (for use in a legend in our GUI) as well as writing the constructor and most of the accessor methods.

Timeline

Week 1 (to Checkpoint 1):

- Create a makefile so that we can compile in one step
- Complete training and data file saving/reading

- Basic text editor with basic functions
    - At this, indication of where to save/load files is still text only
- Start Viterbi algorithm
- Allow parsing by basic separators

Week 2 (to Checkpoint 2):

- File selectors or saving/loading files
- Improved aesthetics of GUI
- Error handling for opening files / file formatting
    - Deal with when files are in the wrong format
    - Deal with when files don't exist or aren't readable
- Complete/debug Viterbi algorithm
- Basic syntax highlighting
- Add exceptions for parsing

Week 3 (to Final Product):

- Debug debug debug
- Add extra features
- More time/space efficient

Version Control

github address: git@github.com:billyjanitsch/CS-51-Final-Project.git