

# Capstone I: Final Report

Tyler Baylson, Andrew Sexton

Spring, 2020

## 1 Initial Progress

In our proposal, we clarified how important it is that users can easily and efficiently control access to their data, starting from simple and straightforward privacy purposes all the way to fully customized schedules of access to highly confidential data. So, though we were only concerned with more simple inputs from the start, code was immediately written to validate initial inputs and run initial input tests. Once we got into the 'flow' of Python, we found it rather intuitive.

As was determined to be a potential best course of action during our first meeting, the spaCy python module was quickly implemented for handling grammar and did end up being extremely useful. It even gave us ideas for entity recognition training and development on Capstone II.

Soon after we started, we built Github processes and workflows, which seemed to be helpful especially for Tyler. He had not had good experiences with branching or merging in the past, and Andrew had a bit more experience so was able to guide. We both learned much from this process, as Andrew noticed some weaknesses in git usage he had not previously been aware of.

Afterwards, careful review of functional intents and test designs during both formal and pair meetings helped establish further division of responsibility and helped us both play to our strengths.

## 2 Basic Implementations

Code was then written to process a given input for primary grammar analysis. At this stage we also began work on a "human readable" series of columns that would show the parsing of data. This came in handy during our final presentation when our primary code did not

operate as expected, but we were still able to show that the program was picking up on specific access triggers and negation.

Test sentences were written for further ideas and we began to become aware of some of the linguistic ambiguities that became a fascination for us moving forward in the project. Andrew got to implement some of his further git learning by performing a rebase/cleanup of the master branch, which made his prior commit messages a little more descriptive. These skills would come into play in other projects in his semester.

We both attended the Research Poster Workshop on March 4 and had some good ideas about poster presentation and copy. This turned out to be a great benefit for us when we submitted the poster for review in the department, and we generally feel pretty good about how it turned out.

This was about when the outbreak of Coronavirus began, which shook up all of our classes and set us back in some ways. We were still able to split our tasks up appropriately and did not experience too much disruption as far as Capstone was concerned.

### 3 Pandemic

Though our work was unaffected, we decided to build new routines around pair programming moving forward and found a great benefit toward our immediate next milestone. At this stage, we were able to refine the localized references of directories, files, and file types and were also working toward establishing methods to avoid semantic ambiguities with potential verb/noun/proper noun overlaps (“access” is a verb, but “Access” is a Microsoft Product). We would later note that these tasks would be good for coverage by our custom implementations in Capstone II and look forward to testing out some concepts there.

We began to implement well-formed notation at this stage, refining our output of well-formatted data to a local policy file. We were able to start the program via command line, parse a sentence, and see that the policy was added to the policyfile in the local directory. This was extremely satisfying.

At this point, spaCy’s core dictionary proved more generally useful than WordNet, so we decided to put it in the foreground. We still have the WordNet resources we originally obtained, but the licensing (and usability) is a little more straightforward with spaCy, so we likely will not return to WordNet in the future.

### 4 Flexibility and Responsiveness

We were then able to scan local directories for given file references.

1. If a file exists as given, it is added as a target resource with “name: target”.
2. If a folder exists as given, it will be added as a target resource with “case: targetFolder”.
3. If a filetype is given and files of that type exist in the local directory, “case: targetExtension” will be added to the policy file.

We do note that if a file with a given extension is not present in the immediate directory, it will not be recognized. This path was chosen rather than scanning against a file of 1500 file extensions. In Capstone II’s entity recognition training, however, this will quickly be addressed.

At this stage, we ran into serious time crunch, as many projects and exams came due very suddenly. They were expected, but we were still no less busy to get them all in. The following week, however, we were able to focus on Capstone and made it a point to get some good pair programming time to finish our sprint objectives.

## 5 End of the Semester

We were also, in this timeframe, able to complete and submit our poster after many hours of work. Neither of us are particularly skilled in design, and though we had considered the opportunity to work with more visually-oriented people outside of the project, we were unable to meet them under the circumstances. We were also unable to touch base with the poster workshop leader to ask for feedback at this time, so we simply did our best.

We began working hard at this point on our capstone presentation itself and found visual aids to be a bit more difficult than expected. But we were able to get some good graphs together that we think help illustrate our conceptually tricky parts of our program. We may only need slight modifications to them for Capstone II.

Overall, we are very pleased with our progress in Capstone I and look forward to working on the project further in Capstone II.