NLP Final Project Report

Tobin Yehle and Dasha Pruss

December 10, 2015

1 Components

All of our relevant source code is inside of `Sherlock.java`. We have a main method inside of `Driver.java`, which does all the IO and data wrangling. We made some data structures for storing Stories and Questions, which are inside of `Story.java`, which contains a private Question class as well. Most of this code was auto-generated (equals, toString, etc.).

Our `Sherlock.java` file contains a single public function, processStory, which calls the numerous private methods that we wrote. Our code is very well documented, so if you'd like to read about the specifics of our inner-workings, you can check out our documentation. For instance, the function findBestSentence searches through the sentences in the story and finds the best sentence points-wise for the given question.

2 Team Member Contributions

We pair programmed our entire project; each of us spent roughly equal time programming. If you would like to see the specifics of who edited what, feel free to browse the commit history of our project in our bit bucket repository. To see who last edited each line of each file you can take a look at blame.

3 External Resources

We used Stanford's CoreNLP libraries for all of our needs (tokenization, sentence splitting, part-of-speech tagging, NER, parsing, and coreference resolution).

4 Regrets

We regret spending so much of our time trying to get co-reference resolution to work because it ended up not helping us at all — in fact, it made our f-score worse. We also wish that we had had time to use WordNet to find semantic classes our of noun phrases, as well as using it to find verb synonyms.

5 Successes

The rules that we built, both from the Quarc paper as well as our own original rules, were highly effective at identifying the correct sentence. We were also happy with the way we extracted phrases from the sentences — it was a very simple approach, but surprisingly effective.