Summarizer51
Functionality Checkpoint

Eric Ouyang and Frederick Widjaja
May 1st 2014

Progress

We've successfully implemented our text parser, a Graph system, and the TextRank algorithm for sentence extraction. In the basic tests that we've run so far, our program seems to do a good job summarizing the given text. There seems to be some minor bugs with text parsing. We're looking into tweaking our parsing algorithm to accommodate for special cases.

Problems

After making significant progress in implementing Summarizer51 in OCaml, we ran into huge problems trying to figure out some of the nuances of OCaml. Spending a lot of time trying to debug the code, we decided that porting our code to Java made the most sense. Especially because the recently released Java 8 implements a lot of the functional programming paradigms used in OCaml, we figured that this switch was apt and appropriate.

At first, we found some of the syntax in Java 8 to be strange and foreign, especially coming from OCaml. However, we ultimately saw parallels between the languages and got the hang of the new features in Java.

Teamwork

We made time to meet together in person to work on the final project side-by-side, splitting the work up evenly into isolated modules that interfaced with one another. We found that writing up our specification enabled us to more easily collaborate with one another.

For the bulk of the implementation of TextRank itself, we did partner coding to achieve successfully working code. It was helpful for us to catch each other's bugs as we developed our implementation of TextRank.

Plan

Now that we have completed the TextRank algorithm for sentence extraction, we want to abstract and utilize our existing code to identify keywords within texts.

Additionally, we want to continue tweaking the parameters for the algorithm to get the best summaries.

As we finish up our project, we'd like to polish up our user interface with helpful command-line parameters and help text.

If we still have time, we'd like to implement LexRank and look into ways to combine our results from LexRank with TextRank.