

Paul Ujlaky, James Doyle, Ronny Cuji

Programming Workshop Lab

Dr. Haolin Tang

Semester Project Reflection

For this semester, we were tasked with creating a Java application which takes a topic of articles and compares each article's sentiment, or overall attitude. Through our work, we feel that we have not only met this goal but exceeded it by creating an interactive app which extends the original purpose by allowing users to easily create topics and add articles either by file path or URL. In hindsight, there are some areas we feel could be improved and many lessons which have been learned.

Creating a command-line-interface (CLI) proved to be slightly more complex than originally imagined. This is largely due to how the different user interfaces needed to interact with each other. A possible solution which improves cohesion within the *Interactions* class is to split each different UI into a different class. To allow for code to be reused, these classes could inherit from a base UI class. This is not necessary, since each UI is split into separate methods within the *Interactions* class, however could be useful for maintainability.

A large improvement we took upon ourselves to make to this project was to add an option for users to add articles by direct URLs. This was implemented due to the fact that many users are not familiar with file paths, especially on command-line-interfaces, and it is much easier to copy an article's link and paste it rather than select all of its content. We used the *JSoup* module to allow us to request a web document, parse the content within its body tag, and then subsequently write it to a file. On this topic, another large challenge for us was handling file paths.

As we knew, different operating systems handle file paths differently. MacOS, built off of Unix, handles file paths similar to Linux, while Windows has its own approach. Although mostly developed targeting Windows, we were able to test the application on Linux, MacOS and Windows, thanks to WSL, and a group member who had a Macbook. Through testing for each platform, many file-path related bugs needed to be addressed. This was a good experience for getting to understand how Java handles this very universal topic in programming. By better understanding the *java.nio* (new io) package, we were able to implement satisfactory solutions such as path resolvers to ensure file paths are handled correctly on all platforms.

As mentioned, the fundamental functionality of this app is to analyze an article's sentiment. Although complicated, this task was achieved using a relatively simple method by taking a list of predefined words with sentiment rankings associated with them (the lexicon dataset) and matching the cleaned article's words against this list. When matching, the sentiment rank of each word was added to a global incrementer, which then was also affected by negative words. This, although effective, could be done in a more practical way as certain articles may not be represented correctly. An on-device machine learning model would be apt to do this task.

In conclusion, this project was a very good way to learn how to create, manage, refactor and implement new features within an actual codebase in Java. Refactoring many times, our source code is totally different than when we started... and even almost entirely differently structured from milestone #1. This was by design, as many lessons were learned through this process, and a very helpful trait of Java (and object oriented programming) is the ability to keep the functionality of existing code and easily encapsulate it in a new class. Even with the areas that could be improved, we are confident that this software and our iterative development of it exceeds the goals initially presented.