Final Project – Text Analysis

by Haoyu Xiong and Iris Fu

Part I - User Manual

As pictured above, the program comprised of 5 buttons which declared and contained 5 different part of the program.

* Upload (Allows the user to upload files for subsequent analysis)
* Characteristic Info (User inputs the author, genre, year and topic for specific file)
* Text Filters (Different options and buttons allow the user to apply certain text filters to specific files, depending on the user’s preferences)
* Statistics (Allows the user to see some basic details and stats of inputted files, including Top and bottom n referenced word, how many words, how many characters, how many lines, and if applicable, the characteristics inputted by the user)
* Prediction (Allows the user to create a decision processing procedure for the program by analyzing the inputted characteristics with ID3 or PCA logics, and after trained with data, this part will allow the program to guess certain characteristics based on the procedure it created)

One thing important for the user to know is that these buttons don’t work separately. That being said, the program will require the user to go from upload first to the predict step by step due to the fact that the user will want the outcome of the program to be as precise and comprehensive as possible, therefore the program with a fixed direction will allow the program to not run on every single operation such as clicking and inputting, which slows down the program and occupy the spaces unnecessarily.

1. Opening Program

The program, after the user ran the interface, will initialize with a frame that contains the five buttons mentioned above and a single instruction line that informed the user to go upload files first.

1. Upload

The upload contains two frame. One of them deals with the uploading and removing files, which contains the entry allowing the user to put in files and the upload and remove button. The two buttons are put together to optimize the program that the user will have choices in case a mistake happened.

The second frame is simply printing the filename that was being inputted. This feature is created to provide the user with a sense of which was being putted and which isn’t.

After the user inputted the files they want, the program will take the string and the user should be able to go to the next step.

One nice thing about the program is that the user can always go back and change what they putted in so that they had the choice not to restart the program.

1. Characteristics Info

This button leads to a big frame with four separated buttons which allows the user to input the four different characteristics we used to analyze the file, including the author, the genre, the year and the topic.

Under the four buttons are two frames, one of which is for the purpose of inputting. If the user already inputted the filenames, this frame will allow them to check the files and add the information accordingly. The idea of checking files is to allow the user to be able to input the same info that two or more files may share. Note that this frame changes according to the button the user clicks.

Another frame is basically a chart that, like the one within the upload, enables the user to visually see and supervise what they have putted in. If any change is needed, add can overwrite what was being inputted.

1. Text Filters

This button leads to three different frames aligned with each other.

The first one is files frame which enables the user to check the files they want to apply with filter.

The second frame is the text filters frame that allows the user to check the specific filters they want, depending on what kind of character they care. This frame contains the button that the user clicks to command the program to apply the filters.

The third one is a state that displays if the filters are applied.

*This feature takes the longest time compared to the other features because it needs to run through every word in the files.*

1. Statistics

This frame contains three parts.

Display Statistics will display the basic stats entered and calculated from the file uploaded as a table.

TopN and BottomN will let user to set up the number of words they wish to have and information will be plotted in separate graphs for every file with the N words and their frequencies accordingly.

1. Prediction

This frame contains ID3 the decision tree and the PCA. Under both frames, user can choose uploaded files to train the tree. In ID3, user also needs to select a classifier, then click Apply, the status message will be displayed at the right. Then the user can click on the Evaluation button on the left and enter the attributes to get the most likely author. In PCA, user can enter the number of words to be considered and select the file to train the tree by clicking Train. Then the user can select the file for Evaluation. A scatter plot will be pop out and the result will be display as the sequence of selected document (top to bottom) at the lower right corner under the Evaluation button.

**Important Note**

PLEASE CLICK on 3(characteristic info) and 4 (text filter) button on row 1 before clicking on the 5(statistics).

Part II – Reflection

In this project, we mainly worked on three parts: team working on a long project, GUI and application of concepts we learned in class.

During this semester-long project, we are trying to work through the checkpoints to accomplish the final text processor. I think this is a very useful experience since the final project in 203 is much easier so we can finish at the end without much planning ahead. Checkpoints are great opportunities for us to make sure that we’re keeping up with the project since there’s no way for us to finish the whole thing at a very short period. However, some elements in the checkpoints are not contributing towards our final product. Team working is still very challenging, since everyone has their own naming for variables and it’s not as easy to understand what others are writing when the program gets long.

We have never been taught how to write a GUI before, so we need to figure it out by ourselves using online resources and tutorials. This helps us to learn how to get useful information from untraditional way (ie. Learning from lectures). In this way, we can learn things even outside of the class and without anyone present to help us which might be the common case for the future.

Also, we applied what we’ve learned in the class to this project. For example, we used Heap for the topN method. We need to familiarize ourselves with the materials in class and understand how to write them out in code, and furthermore apply them. Therefore, concepts are not something only exists on PowerPoints but a concrete thing we have worked on.

Reflection for Haoyu Xiong

This project for us was not just a program that we are trying to build. It’s more like a self-learning process combined with what was taught in the class. The checkpoints separated the projects into parts, which really helps us with understanding the materials and applying them in a practical level.

For specific part of learning outcome, I think as a whole, the application of different algorithm and data structures helped me understand the advantages and the disadvantage of different ones and the reasons of using certain structure instead of the others.

One thing I really enjoyed and learned is the team cooperation. Me and my partner communicate very well into how things should be implemented, and how our separate contributions can merge comprehensively. That’s definitely an experience that I can apply into both my career and my daily life.

As for the difficulties, I believe that we stumbled a lot into the debugging part. The communication I had with my partner helped us make sure that we know what to do. But the same as the daily coding, bug happens. And debugging definitely takes more time and effort because more of time it’s trivial.