Sakshi Haria, [Liliana Reyes Medrano](mailto:lreyesmedrano1@babson.edu), [Raphael Dunand](mailto:rdunand1@babson.edu)

MIS3640

Professor Zhi Li

3/17/2021

**# Project reflection and Write Up**

**# Project OverView**

In our project, we compared two literary works from the U.S and U.K in the 18th, 19th, and 20th centuries. These six pieces were retrieved from Project Gutenberg. The following books were analyzed:

1. \_18th century\_

1. UK - A Modest Proposal

1. US - Hannah Webster Foster

1. \_19th century\_

1. UK - Frankenstein

1. US - Cape Code

1. \_20th century\_

1. UK - Virginia Woolf

1. US - The Great Gatsby

In order to analyze the text, we decided to use the following techniques to learn more about textual styles and the similarities and differences that each country had during the designated century: Characterizing by word frequency, text similarity and differences.

**# Implementation**

Once we selected books as a datasource, we choose 6 different books based on their country and date of publication. Once these books were chosen, they had to be imported into python. To do this, we created a function that would import the book from the url and save it as a “book name”.txt file in a folder called data. This would make it easier to analyze the books as they would already be saved in the repository. After the data was imported, we were able to create functions to analyze the text. The function first removes the header and the footer of the book --amount of words and unique words. In the Python file, we broke each analysis into different steps that would resemble an inverted triangle. We started the process by focusing on the story itself--without the header-- and collected all the words that each book has, Then were narrowed down the search for finding the unique/different words. Having these in sequence would allow us to use the previous functions efficiently.

**# Results**

Something that interested us are the similarities that we found through the texts and the language that they shared

|  |  |  |  |
| --- | --- | --- | --- |
| Book | Amount of words | Unique words | Amount of words per unique word |
| The Coquette by Hannah Webster Foster (US 18th century) | 59230 | 5981 | 9.90 |
| A Modest Proposal by Jonathan Swift (UK 18th century) | 3423 | 1089 | 3.14 |
| Cape Cod by Henry David Thoreau (US 19th century) | 79600 | 8529 | 9.33 |
| Frankenstein by Mary Wollstonecraft (Godwin) Shelley (UK 19th century) | 75279 | 7050 | 10.68 |
| The Great Gatsby by F. Scott Fitzgerald (US 20th century) | 48836 | 5959 | 8.20 |
| Mrs Dalloway in Bond Street, by Virginia Woolf (UK 18th century) | 3148 | 1019 | 3.09 |

**# Reflection**

From a process point of view, as a team, we managed to communicate with each other in order to make sure we understood what was going on. We were able to meet up and discuss what needed to be done. One thing that we are able to improve would be managing our time more efficiently. We had a couple of instances where we had delayed or canceled meetings which would have served beneficial for us to make more progress in our project.

Based on how we were working, the team ended up splitting the work into three different sections from data collection, constructing functions, and reflection. We were relatively on schedule on this but there is always room for improvement in order to provide more support for each other. Going into the project, on a more basic level. We wish that we had a little more practice with sharing documents and editing each other’s work. We spent some time figuring this out and if we would have known what to do from the beginning we would have been able to save some time as well and work on the code together more effectively.

Issues that came up were mostly about the meetings. Since time is scarce for everyone, rescheduling and/or miscommunication happened over the weekend. We just had to reschedule and do our best to respect everyone's time commitments. As to what we would do differently. We might have created more frequent smaller sessions to work on the code together. That way no one person would have more knowledge on what was going on in the project than another. This would have not only helped with everyone's understanding but it would help to support each other’s different levels of understanding Python.