# Deliverable 2: Public Domain Book Dating

Garrett Kinman

## 1 Problem Statement

Language changes with time. This is readily apparent to any human reader that picks up different books from several different centuries, but getting a machine to recognize this is much more challenging. To approach this problem, this project aims to predict the age of a public domain book by analysis of the language of the text itself.

## 2 Data Preprocessing

The dataset for this project is Project Gutenberg, an online source of free etexts for over 60,000 public domain books. In the initial proposal, the website was going to be scraped for English etexts available in plain text and with Library of Congress numbers available (to scrape LoC for publication dates). However, there were several complications with this. First of all, the website itself does not permit scraping (a message in the HTML of the pages made this fact clear). Instead, it redirects would-be scrapers to a page where the entire collection can be downloaded at once. However, due to the sheer size of the download (according to one user, over 11 GB and three-and-a-half hours to download), attempting the download crashed Google Colab on several occasions.

As a result of these difficulties, I manually downloaded a very small sample (n = 10) of books meeting the above criteria so as to be able to devise the basic preprocessing algorithms and machine learning models. This is purely a holdover until I can succeed in downloading and extracting all the necessary data from the dataset.

In terms of preprocessing, however, there was also a significant amount of work. While the etexts are fairly standardized in formatting, there are numerous special cases (many of which have yet to be addressed, as they will likely only become visible with a larger dataset). The basics of it so far have been to clean the entire texts by cutting off Project Gutenberg's extra information added at the beginning and end of the files, setting everything to lowercase ASCII characters, removing newlines and extraneous whitespace, and setting classifier values based upon the publication date. For the limited sample, this was limited to century classification, including the 1700s, 1800s, and 1900s. Finally, the texts were vectorized to the most common word n-grams. Given the limited dataset at the moment, the 'n-gram-ulary' size is very much liable to change. Likewise, the *n* size in n-gram is possible to change, as well, from its current value of three (i.e., collections of three words in a row).

## 3 Machine Learning Model

Due to the incomplete state of the dataset, specifics of the machine learning model were not decided. However, due to the long length of the books (tens of thousands of words => tens of thousands of n-grams), the model is liable to be computationally intensive. Thus, the convolutional neural network will be created using TensorFlow, which offers GPU acceleration support.

## 4 Preliminary Results

At present, there are no preliminary results to report, again a result of the incomplete dataset.

## 5 Next Steps

The natural next step from this point is to download and create the dataset from the downloaded etext library. After this, tweaks on preprocessing will likely be necessary, as a wider array of books will increase the number of special cases to handle, such as accented characters, which are currently not handled. Especially due to the goal of this project, for many of these special cases it would be valuable to treat carefully, as use of accented characters or certain types of punctuation or usage of contractions could all be valuable indicators of age.

As part of preprocessing, another step could be to cut out more of the forewords and afterwords and other such extraneous materials from the texts. This is because many of these are written long after the original work, and might negatively impact the model. Clearing out these inconsistent materials would likely require the use of regular expressions.

After these, it will be necessary to run the code on a local machine, rather than Google Colab. This is because of the size of the download, which has repeatedly crashed the webpage when attempting to download it through Google Colab. Additionally, running it on a local runtime would ensure the files stay where they are, so no chance of 11+ GB of dataset disappearing with a browser crash.

Also, once the dataset is downloaded, it will be necessary to read through the metadata for Library of Congress numbers (some books have them listed, and some do not). For those that do, the Library of Congress website will be scraped simply for the publication dates.

Finally, the model will have to be implemented, trained, and tested.