Authorship Identification

Beau Carlborg, Clara Richter, Sunny Chiu

Abstract

Authorship identification is the problem of determining the authorship of an anonymous text based on their previous written work. The authors of this text implemented a fully functional data pipeline consisting of web scraping retrieval, basic natural language data processing, as well as supervised classification model training and evaluation. Precisely, we look into the problem of, identifying the author among a set of candidates given an excerpt of English text. Specifically, we looked into 10 English authors of the classical era. The constructed models use stylometric features -- quantifiable features that capture some element of a work's style -- to perform the classification. Using the modeling techniques of Logistic Regression and Support Vector Machines and k-fold cross validation, the models achieved an accuracy ranging from 90% to 8% depending on the number of authors being classified as well as the number of author style feature traits trained on.

Introduction

As the title alludes to, the problem the authors seek to tackle is that of authorship identification. Authorship identification is widely defined as the process of identifying the most probable author for a book given an excerpt from a work that can be used to extract features from the text. These extracted feature allow us to identify stylometric elements in the text that are indicative of a particular author's style. It is the hope that if an adequate amount of data is collected and effective features are chosen, our constructed model will be able to correctly identify the correct author of a text more often than baseline such as that of the proverbial monkey, which picks authors at random.

The problem of authorship identification falls under the general category of a discrete supervised classification machine learning problem. An extensive amount of research exists on varying problems of classification in the field of Machine Learning and Data Science in general. As a result of this, there are a wide variety of possible algorithms and models which we have available to us. We chose to utilize many of these resources and libraries[1] which had already been constructed for machine learning purposes.

[1] http://scikit-learn.org/stable/supervised_learning.html#supervised-learning

Data Acquisition

Project Gutenberg (gutenberg.org) was our primary data source for books of various authors. Project Gutenberg provides access to over 50,000 books in many formats (plain text, html, and other e-book formats). The project was started in 1971 as an effort to digitalize the books and cultural works and in order to "encourage the creation and distribution of e-books"[2]. Anyone is free to transcribe a book and upload it to the website, as long as the copyright limitations of a given work are respected.

Our goal is to retrieve as many texts as possible from gutenberg.org for each author of our list. It was crucial for this task to understand how the website is organized. On gutenberg.org, each work has a unique id number, assigned in the relative order in which the book was transcribed added to the website. The link to any given book is simply given by http://www.gutenberg.org/ebooks/<id number>, on which links to specific text formats are given. To retrieve relevant texts, we thus had to find the respective id numbers. We considered many different methods for obtaining the books from the Gutenberg site. Eventually, we decided to use a comprehensive catalog of all 56,000 book on the Gutenberg site written in a plain text file format. In this catalog, each book is listed along with its author, book id and other information such as language and subtitles. A standard entry in the list looks like the following:

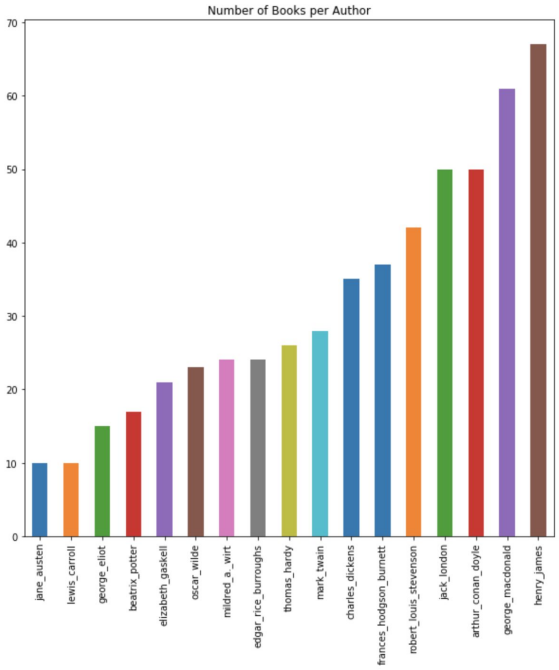
```
The Barbarity of Circumcision as a Remedy for Congenital          57083\n
      Abnormality, by Herbert Snow\n
\n
Alice's Adventures in Wonderland                                  57084\n
      Abnormality, by Herbert Snow\n
```

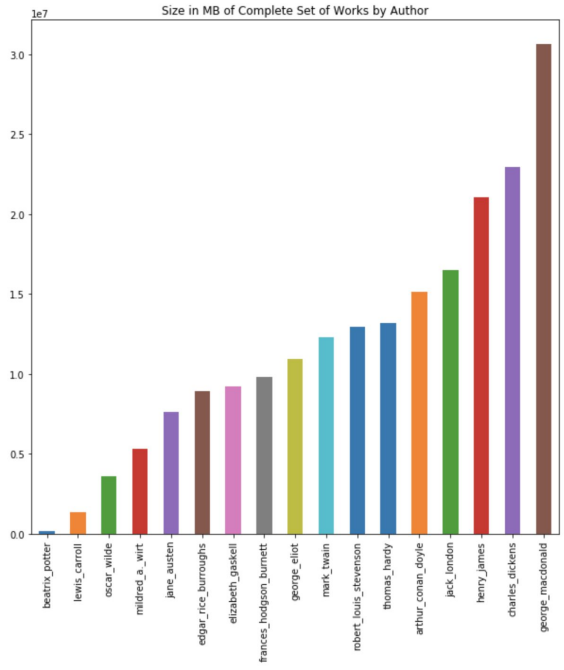
(note: any two entries are generally separated by one or two newline character \n)[3]

From this catalog we wanted to create a uniform and organized dataset in tabular form of every books relevant to our project. To achieve this task, we used regular expressions to find relevant entries and parse out their information. The parsing proved to be a much greater challenge than we initially expected. The unexpected difficulty arose out of the formatting of the list. As it turned out, the books were listed in an extremely non-uniform manner; there was little consistency in the formatting and structuring of the book entries in the catalogue. One particularly difficult part of the process was encountering books written in non-English languages. Oftentimes the texts were listed with proper tags indicating its written language; however, the few examples of books not listed with proper tags proved difficult filter out

[2] https://www.gutenberg.org/wiki/Gutenberg:Project_Gutenberg_Mission_Statement_by_Michael_Hart
[3] http://www.gutenberg.org/dirs/GUTINDEX.ALL

correctly. In the end, we were able to retrieve the relevant information of 20-50 books for each author in our list. After refining our regex parser and webscraper, retrieving the plain texts was a trivial process.





Preprocessing

Data Cleaning

The first step of the preprocessing was to remove the non-english books that our parser failed to filter from the data set manually. Our second step was to standardize the formats of all books. As stated in the introduction, our goal was to analyze texts, and classify authorship, based on their textual content using metrics such as word frequency, word richness, and sentence structure. Thus, it was necessary to take out any overhead content of the text files that were not included in original publication or not intended by the author. In particular, each book had Gutenberg licensing and copyright agreements at the beginning and end of the file. Most books also had a section of “Transcriber’s notes” at the end of the text, and other text chunks that we thought would be irrelevant (or harmful) to our content analysis. Some of these irrelevant data chunks included the table of contents (not indicative of the writing style of an author) and prefaces (in many cases not written by the author). It could be argued that table of contents actually could be useful in identifying an author. For example, maybe some authors always include table of contents, while others never do so. Moreover, from table of contents, the

number of chapters, length of chapters, and general structure of a book might be deduced. However, since we sought to analyze texts based on their the lexical and syntactical features rather than their structural features, we chose to exclude table of contents from analysis.

We discussed several ways of going about stripping the text files. While the prospect of writing a script to automate the process of stripped irrelevant data from the text was alluring, we chose instead to strip the texts manually. This was primarily because there was no strong and uniform pattern in the elements of the text we sought to remove. Initially we thought that we would get around this problem by finding the maximum length of an irrelevant text chunk, and then locate the start of the licensing text, and then simply take out a given number of lines below for each book. However, we decided against this idea for two reasons. Firstly, finding the maximum size of the header data would have required us to manually scan all the books regardless. Secondly, since our dataset was already relatively limited, we decided that it would be wise to retain as much information as possible in the the texts. Stripping the texts manually enabled us to take out texts chunk with human supervision. This approach came with the additional benefit of providing us with the chance to gain an intuition for type textual content in each book.

During this process, we observed that in many of the raw text files, countless words were (for some inexplicable reason) surrounded by underscores or dashes (in one of the books of Elizabeth Gaskell, a single word was surrounded by more than 10 underscores on each side, making it into an apparent 40 character word -- clearly, this was not intended by the author). The elimination of underscores and dashes was achieved with simple regular expressions. This method came with a drawback however. Dashes are in many cases part of the actual text and very intentional. Unfortunately, our regular expressions didn't account for these cases, but took them out altogether.

Feature extraction

A set of predefined features is needed in order to classify data using supervised classification algorithms. In authorship identification, we seek out features which are style markers that quantify an author's writing style. We invested a few days of our projects work into reviewing prior research done on author identification to get a sense for the features that others had used to create strong authorship identification models. One particular article describing the problem of, and previous research on, authorship identification provided us with a comprehensive list of possible features to use when constructing a model. The author identified three groups of style markers that have been used for authorship identification: lexical features, syntactic features, and structural features. *Lexical features* parse a text as a sequence of word tokens. Features that can be measured include word length distribution, sentence length distribution, vocabulary richness, use function words, and n-grams. *Syntactic features* capture the usage of parts of speech and sentence structure. The author writes that parts of speech are a good metric for quantifying writing style since their specific usage is "often unconscious[...] and patterns for a specific author can emerge given enough text" [2]. Lastly, structural features

represents how an author “arranges the layout of the document”, including the lengths of paragraphs and chapters. For classifying the author of a book, we decided to use lexical and syntactic features. We chose not to use structural features of a text because of the wide variety of formats which project Gutenberg books are written in. This variation among books is primarily due to the autonomy that transcribers have when re-writing. They are able to write the text version of the book in any way they see fit; and therefore, we did not believe that the author's style was captured by the structural features. The features that we decided to measure in the end were as follows:

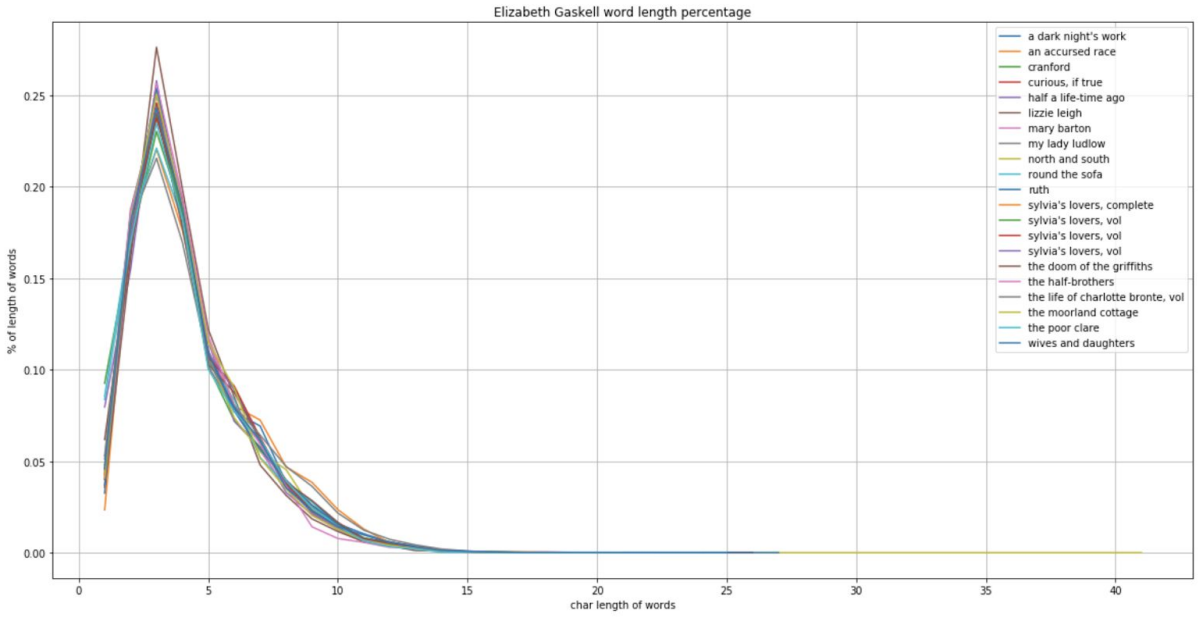
Lexical features

1. word n-grams (up to length 3)
2. word richness[4]
3. length of words (mean and standard deviation)
4. length of sentences (mean and standard deviation)

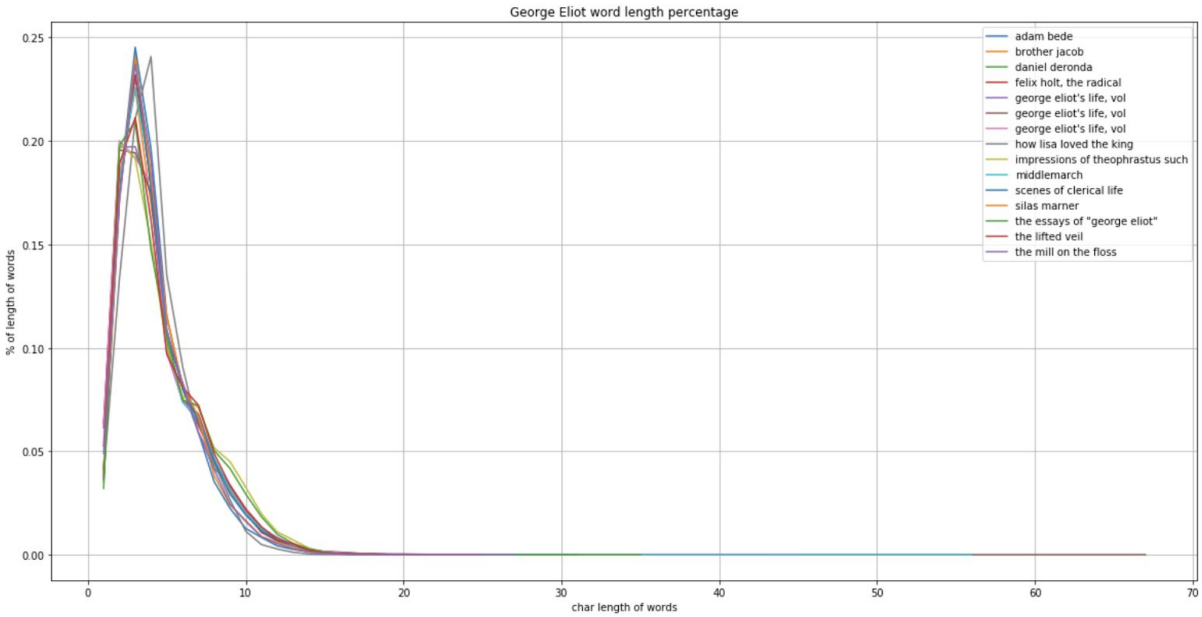
Syntactic features

5. parts of speech n-grams (up to length 3)

It is worth noting that the dimensionality was rapidly increased with the incorporation of word and parts of speech n-grams. In fact, using n-grams of length three gave a dimensionality of 300,000 for one single book. It easy to see that combining the n-grams of several books would increase this number by a significant factor, giving us potentially millions of features. A dataset with this many dimensions would clearly be infeasible to train on and would not offer large enough improvements in the model’s performance to warrant the runtime. Moreover, there is a high likelihood that the resulting model would be overfit to the specific works we provided in



ıg



our training set. We put a lot of effort into finding intelligent ways to decrease the dimensionality. We decided on two different criteria any column had to satisfy in order to be kept. Firstly, the feature represented by the column needs to occur in some constant percent of the works we train on. This is crucial because as our number of n-grams and parts of speech grams increases, there are many feature for which a statistic can only be retrieved from one work. These columns do not help identify an overall author's style, and therefore we remove them. In addition, we removed columns when the summary statistics retrieved from the data for each author are too similar. This allows us to reduce the number of columns for which there is only a slight difference in the feature's presence in the author's works. The dimensionality reduction finalized our data preprocessing step.

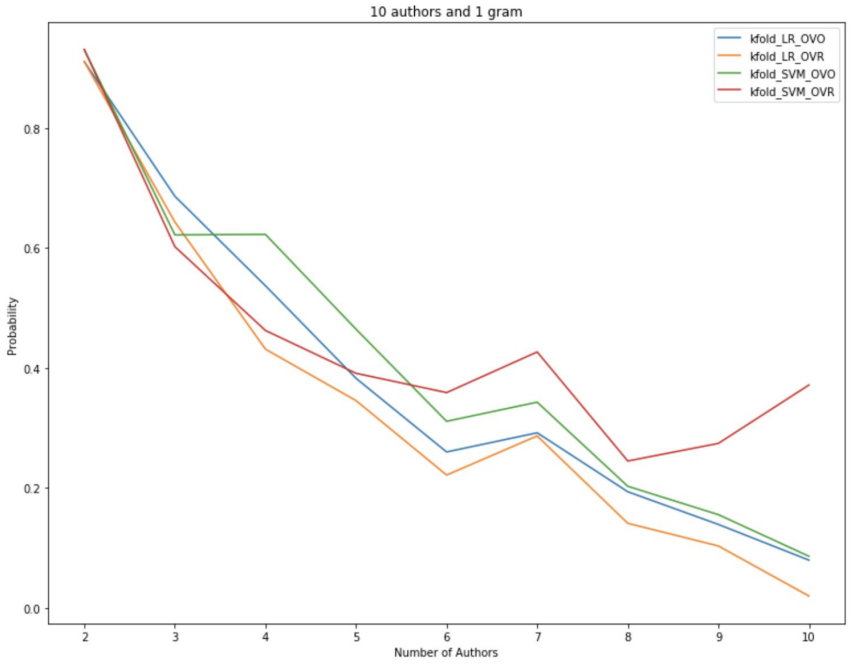
Modeling and Prediction

As stated in the introduction, our problem was one of discrete supervised classification; we sought to identify the probabilities of a work being written by each in a set of candidate authors. The nature of our problem ruled out a few modeling techniques more apt for regression problems (such as linear regression), and unsupervised learning (clustering). We thought that several models would potentially work well for our classification. In the articles we read the techniques most frequently used include Support Vector Machines, Artificial Neural Networks, k-Nearest Neighbour, Random Forests, and Decision Trees. In [2], which looked authorship identification for the specific domain of classifying research papers, they found that SVMs and ANNs reached the highest accuracy out of these. At the end, we decided to go with two modeling techniques: logistic regression and SVMs. While it would have been interesting to try more than these two, we decided to limit ourselves because of time constraint. Logistic regression is one of the simpler techniques, and we figured that it would be good to start with.

SVMs are closely related to logistic regression but more sophisticated. For the logistic regression and support vector machine, we performed both binary, one versus one, classification and one versus many, when we increased the size of the set of possible authors to choose from. We performed 10 fold cross validation for each model.

Results

We observed that 1 gram fairly high prediction accuracy compared to 2 and 3 grams. In addition with the long computation time to retrieve 3 grams, we decided to retrieve 1 grams from 10 authors. As seen in the figure below, there is a downward trend of accuracy for all models except one vs. rest support vector machine. The one vs. one support vector machine had the highest accuracy for author classification of 2 to 5 authors while one vs. rest support vector machine had the highest accuracy for author classification to 5 to 10 authors. After 8 author classification, one vs. rest support vector machine there is an upward trend. Larger author classification samples are required to observe the trend of the one vs. rest support vector machine.



Issues Encountered

1) The difficulty of data acquisition -- why we changed to authorship identification
2) Working with big data and the need for efficiency:
   i) Appending to dataframe is slow -- have to reallocate space for a whole new dataframe whereas appending to dictionary is fast

Future Work

Most of our work was spent on data cleaning and preprocessing which left a limited amount of time for predicting and modeling. Going forward with the project, we would explore more models. Specifically, we think that k-nearest neighbour and artificial neural networks would possibly yield good results. Sci-kit learn provided a very friendly user interface, and didn't really challenge us to understand the algorithms perfectly before applying them. In an ideal world, we would implement the algorithms ourselves. While doing so we would gain a deeper understanding for why some models are more apt to tackle are problem than others. We would also like to enhance our domain knowledge, so as to be able to extract and refine features in a way that better represents author writing style. As mentioned in the feature extraction section, there were several stylometric features that we omitted from our analysis. These include more precise metrics of so called "function words" and punctuation. In [2], it is reported that these are features that have been used extensively in previous research.

Summary and Conclusion

Our project represented a significant amount of work attempting to collect, clean, and process our data. From the onset, we knew our project had a data set with an enormous amount of potential, however it would require a significant amount of data munging. In order format the data. As we began working with our data, we observed many difficulties in our process of constructing an array of features which we could use to identify and authors work. After much deliberation, we decided to use a set of basic lexical features as well as a set of N-Grams and Parts of Speech N-Grams. After we processed our data, we began run the models using our various features we constructed. We chose to use a logistic regression and a support vector machine. When running our models, we observed many interesting results. Perhaps the most interesting results were observed as we saw the scaling of authors and N-Grams. As the Number of authors in our training and Data increased, we observed worse results for each classification. Contrary to our initial expectations, we also observed, that as the number of Grams increased in each of our data training sets, we also observed a decrease in the success of our model. We believe this is a result of overfitting in our model. All said and done, we are very pleased with the overall outcome of our model. While we invested a very large amount of time into the process of learning how to create stylometric features, we believe this paid off immensely and resulted in a much stronger model.

Cited works

[1] https://brage.bibsys.no/xmlui/bitstream/handle/11250/2353615/12344_FULLTEXT.pdf?sequence=1

[2] https://web.stanford.edu/class/cs224n/reports/2760185.pdf

[3] http://www.gutenberg.org/