TEXT MINING AND AUTHOR PREDICTION

BIRINGA CHIDERA, ANIRUDH REDDY SURAM, RICHARD DONBOSCO, VIKAS JAISWAL

Abstract. Authorship prediction refers to the association of an author to a specific document. In literature, it is very important to know about historical text who wrote it and when it was written. However, we can also predict the nationality of the author, characteristic styles of the author and genre of the text. In recent years, there are lots of classification and prediction techniques conducted in this process. In this paper, we explore the problem of Authorship Prediction using Text Mining and Sentiment Analysis. This analysis predicts the authors based on a collection of text derived from their authored books. We have performed experiments on books that are extracted from Victorian Era Authors datasets by using different features such as bag of words, n-grams or TF-IDF models. We also implemented different classifier techniques to improve our success rates. The techniques discussed in this paper is a two stage process, where in the first stage, text mining and sentiment analysis are performed and in the second stage different classification models are trained to predict the authors.

1. Background and Introduction

Literature text mining and authorship prediction have always been a problem in the field of data mining, due to major demand in computational resources and ambiguous text set. Authorship prediction is the process of attempting to identify the likely authorship of a given document, given a collection of documents whose authorship is known [1]. In recent years, authorship prediction becomes an important problem as with the fast-growing Internet usage increases range of anonymous information increases. Any historical text discovered can be used to know about the original author, nationality of the author, the characteristic styles of the author, genre of the text. However, the problem of author identification for the general public is made challenging by the differences between social media posts and traditional forms of writing such as books, newspaper articles, and research papers. It is useful when two or more people claim to have written something or when no one is willing (or able) to state that she or he wrote the piece. Authorship prediction of online documents is different from the authorship prediction of traditional work in two ways. Firstly, the online documents or text collection is mostly unstructured, informal and not necessarily grammatically correct as compared to literature, poems and phrases which are syntactically correct, very well structured and elaborative in nature. Secondly, for a single online document the number of authorship disputes are far more as compared to traditional published documents, that is because one of the challenges with authorship prediction in this case is scarcity of standardised data to test the accuracy of results [2]. The goal of Authorship prediction is to identify authors of texts through features derived from the style of their writing; this

Date: December 18, 2019.

is called Stylometry. Applications of authorship prediction include plagiarism detection (e.g. college essays), deducing the writer of inappropriate communications that were sent anonymously or under a pseudonym (e.g. threatening or harassing e-mails), as well as resolving historical questions of unclear or disputed authorship [3]. In this paper, we provide comprehensive stylometric technique for contextual and stylistic extraction for literature classification and prediction based on certain identifying features, such as the author and genre of the text. We have performed text mining and sentiment analysis on texts that are extracted from Victorian Era Authorship Attribution dataset by using different features such as bag of words, TF-IDF, n-grams. We performed experiments on different features extracted from these texts with different classifiers technique and combined these results to improve our accuracy rates. According to conducted experiments, the success rates dramatically changes with different combinations, however the best technique among them is Random Forest. We also have aimed to provide suitable technique for author prediction research community and demonstrate how it can be used to extract features in any kind of authorship problem.

2. Related Work

In the field of authorship prediction, hundreds of researches were conducted in the last 10 years. Stylometry, linguistic characteristics of a language were studied to gain knowledge about the author of text. With the increasing number of documents in Internet, and as most of the writings are anonymous, authorship attribution becomes important. The researchers are focused on different properties of texts. There are two different properties of the texts that are used in classification: the content of the text and the style of the author [4]. According to last researches in 2001, Stamatatos, Fakotakis, Kokkinakis [5] have measured a success rate of 65% and 72% in their study for authorship recognition, which is an implementation of Multiple Regression and Discriminant Analysis. Also, in 2003, Joachim Diederich and his collaborators conducted experiments with support vector classifiers and detected author with 60-80% success rates with different parameters [6]. R. Rousa Silva et.al [7] have worked on authorship attribution using stylistic markers for tweets written in Portuguese. Their analysis shows how emoticons and short messages specific features dominate over traditional stylometric features to determine the authorship of tweets. The final results show significant success (i.e. F Score = 0.63) for 100 examples available from each author under consideration. Bhargava et al [2] combined lexical, syntactical, tweet-specific, and metadata features with an SVM classifier and a RBF kernel to obtain high precision (up to 95%) with a higher number of authors (10 to 20). Anderson Rocha et al [8] produced a paper, the main ideas of the paper were to develop techniques that will inform the direction of the investigation by identifying authors and also create adaptable authorship attribution to each group of users and social media environments. Most of the studies are conducted with one or two classifiers and with limited feature sets. We do not know a comprehensive study in this field. Our study differs from others by conducting many tests with various feature sets and classifiers.

3. Problem Definition

The purpose of this study is:

- To perform a robust text mining analysis using feature extraction techniques
- To extract insight from sentiment analysis
- To accurately classify authors based on their text

4. Methodology

In this section, we will explain our project steps. Text Mining and Authorship Prediction process consists of gathering texts which are the observations to be classified in some sense, a feature extraction mechanism that computes numerical or symbolic information from the observations, doing sentiment analysis, a classification or prediction models that does the actual job of classifying or predicting observations and at the end performing model evaluation based on the predicting models.

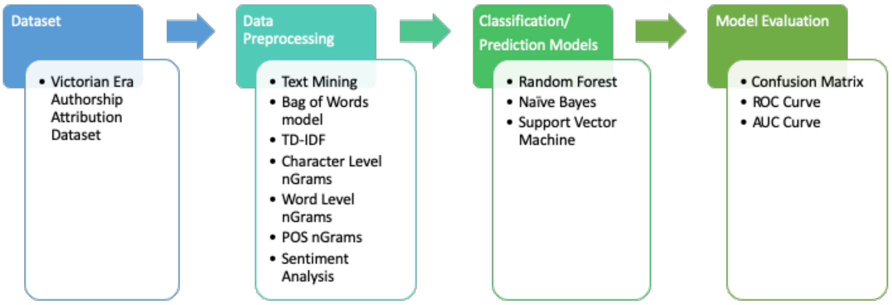


Figure 1. Workflow of the methodology

4.1. **Dataset.** For performing the text mining and prediction, we used Victorian Era Authorship Attribution dataset from UCI Machine Learning Repository. Now, from the largest publicly available authorship attribution dataset, based on lack of sufficient computational resource, we will be sampling our data to include only two well-known Victorian-era authors. Dataset contains 18th and 19th Century English and American Authors Book.

4.2. **Data Processing.** Data Preprocessing is a critical step in filtering out the unstructured data and extract the meaningful information for training and testing our models. We have preprocessed our data in two pipelines: Text Mining and Sentiment Analysis. Text Mining is the process of examining large collections of text documents to discover new information which can be further analyzed such as tokenization, removing numbers, stop words, stemming etc. On the other hand, Sentiment Analysis is the mining of contextual data and classifying opinions as positive, negative or neutral which is very important in understanding the emotions of data.

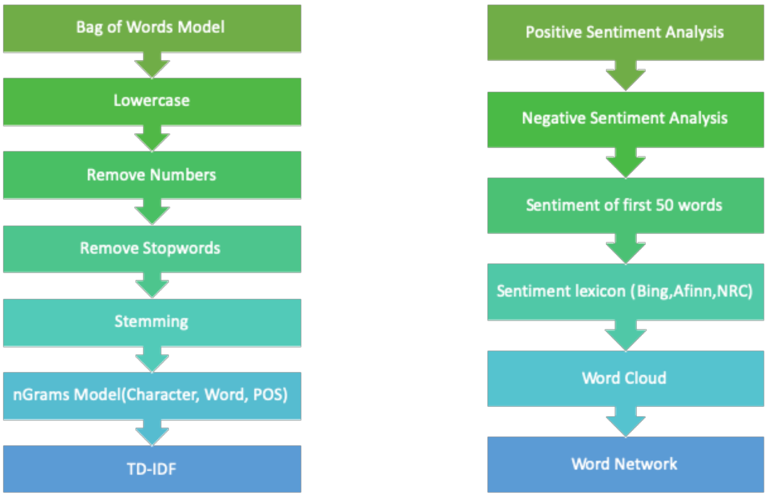

Bag of Words Model
Lowercase
Remove Numbers
Remove Stopwords
Stemming
nGrams Model(Character, Word, POS)
TD-IDF
Positive Sentiment Analysis
Negative Sentiment Analysis
Sentiment of first 50 words
Sentiment lexicon (Bing,Afinn,NRC)
Word Cloud
Word Network

FIGURE 2. Data Preprocessing and feature extraction pipeline for sentiment analysis

4.3. **Features Selection and Extraction.** After filtering the unstructured data and acquiring pure data, another important step is to find distinctive features from this preprocessed data. In authorship prediction, not only the content i.e the text itself are important, but also stylometry and other features that define the characteristics of a writer. We would be discussing in brief about these features

4.3.1. *Bag of Words.* Bag of Words model is one of the important models in simplifying representations used in natural language processing and information retrieval. It is commonly used in methods of document classification where the frequency of occurrence of each word is used as a feature for training a classifier.Let's consider a simple text document: "John likes to watch movies. Mary likes movies too." Representation will be BoW: ”John”:1,”likes”:2,”to”:1,”watch”:1, ”movies”:2. After transforming the text into a ”bag of words”, we can calculate various measures to characterize the text. The most common type of characteristics, or features calculated from the Bag-of-words model is term frequency, namely, the number of times a term appears in the text.

4.3.2. *TF-IDF.* TF-IDF is an information retrieval technique that weighs a term's frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF-IDF weight of that term. TF-IDF algorithm is used to weigh a keyword in any content and assign the importance to that keyword based on the number of times it appears in the document. More importantly, it checks how relevant the keyword is throughout the data, which is referred to as corpus. Term Frequency of a word is the frequency of word (i.e number of times it appears) in a document while Inverse Document Frequency of a word is the measure of how significant that term is in the whole corpus.
$\mathrm{TF(w)} = \frac{Numberoftimestermwappearsindocument}{Totalnumberoftermsindocument}$

IDF(w) = $log_e \frac{Totalnumberofdocuments}{Numberofdocumentswithtermsweinit}$

TF - IDF = TF(w) x IDF(w)

4.3.3. *Word-Level nGrams.* An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a (n-1)-order. Word-level n-grams let us capture more semantically meaningful information from a text in the form of short phrases. n-gram of size 1 is referred to as a ”unigram”; size 2 is a ”bigram” and size 3 is a ”trigram”. The following example shows a simple phrase and the complete lists of unigrams, bigrams, and trigrams extracted from it [8].
Text: “To be, or not to be, that is the question.”
Unigrams: (“To”, “be”, “or”, “not”, “to”, “be”, “that”, “is”, “the”, “question”)
Bigrams: (“BEGIN To”, “To be”, “be or”, “or not”, “not to”, “to be”, “be that”, “that is”, “is the”, “the question”, “question END”)
Trigrams: (“BEGIN To be”, “To be or”, “be or not”, “or not to”, “not to be”, “to be that”, “be that is”, “that is the”, “is the question”, “the question END”)

4.3.4. *Character-Level nGrams.* Character-Level nGrams, where the items consist of one or more characters and we can extract unusual features such as emoticons and special use of punctuation. They are often used in text mining and authorship attribution because they also help to mitigate the effect of small typos that authors do not repeat very often, which are not style markers. For example, for the word “misspelling”, the generated character-level 4-grams would still have “miss,” “issp,” “sspe,” “spel” and “ling” in common with the 4-grams generated for the correct word “misspelling.” This decision improves efficiency by cleaning unstructured data and removing noisy features that are unlikely to appear again in the dataset.

4.3.5. *Part of Speech nGrams.* The simplest stylistic features related to syntactic structure of texts are part-of-speech (POS) n-grams [10]. POS tagging is process of analyzing sequence of words and attaching a category for each word in the sequence. One of the main concerns is presence of noise in data and POS tags can be very helpful in reducing the feature-space to a limited number of very general elements. POS tagger in authorship attribution is that grammatical usage can serve as an important indicator of an author’s style even in short messages. Common tags are “N”, “P”, “V”, “O”, “W”, “A”, and “D” correspond to noun, preposition, verb, pronoun, punctuation mark, adjective, and determiner, respectively

4.3.6. *Stemming.* Stemming is the process of reducing a word to its root/base word. It is important in linguistic studies, artificial intelligence, text mining, information retrieval and extraction. Because of the grammatical reasons, in text documents words coming from same root utilized differently based on their usage as adjectives, adverbs, nouns, tenses of the verbs. For example, “cook”, “cooking” and “cooked” are different forms of same stem of “cook” or “go, went, gone” are different inflicted forms of “go”.

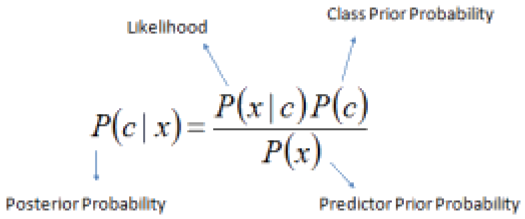
4.3.7. *Sentiment Analysis.* Sentiment Analysis is contextual mining of text and classifying opinions as positive, negative or neutral. Moreover, it has many real-world applications like analyzing survey responses, product reviews, social media comments etc. In our project we performed sentiment analysis on author’s text and also compared

sentiment dictionaries for differences in categorization of sentiments. We also explored Sentiment Lexicons using a tidytext package which provides distinct lexicons whereas these lexicons are dictionaries of words with an assigned sentiment category or value. Tidytext provides three general purpose lexicons:
AFINN: assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment.
Bing: assigns words into positive and negative categories.
NRC: assigns words into one or more of the following ten categories: positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. At the end, visualized word cloud and word network graph among different authors.

4.4. **Classification / Prediction Models.** Various classifiers can be applied to our dataset. We are using Cross-validation technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. We implemented following classifiers to find a best fitting classifier for our dataset.

4.4.1. *Random Forest.* Random Forests is a method which comprises a collection of classification or regression trees, each constructed from a random resampling of the original training set. In addition, it is a classifier that evolves from decision trees. It actually consists of many decision trees. To classify a new instance, each decision tree provides a classification for input data; random forest collects the classifications and chooses the most voted prediction as the result. The input of each tree is sampled data from the original dataset [11]. Random Forest has become a commonly used tool in multiple prediction scenarios due to their high accuracy and ability to handle large features with small samples. Let's consider certain parameters while building trees and splitting nodes in Random Forest. We can write the Entropy equation as: $Entropy(t) = -\Sigma_i = 1^c p(i|t) log_2 p(i|t)$, where p(i—t) fraction of records associated with node t belonging to class i.
Gini Equation as: $Gini(t) = 1 - \Sigma_i^c [p(i|t)]^2$
Information Gain as: $IG = I(parent) - \Sigma_i^k \frac{N(vj)}{N} I(vi)$
We compared the impurity of the parent node with the average impurity of the child nodes. Briefly in Random Forests (both, regression and classification), each estimator in the ensemble is built from a bootstrap sample from the training set. When the algorithm splits a node during the generation of the decision tree, the chosen split is no longer the best split of all the features. Rather, the split that is selected is the best split of a random subset of the features. Due to this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance decreases, usually more than compensating for the increase in bias, finally this produces a better model [12].

4.4.2. *Naïve Bayes.* Naïve Bayes is a classification technique which is based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature [13]. Bayes theorem provides a way of calculating posterior probability $P(c|x) \, from \, P(c)$, $P(x) \, and \, P(x|c)$. We can represent bayes theorem in equation below.


Likelihood
Class Prior Probability
$P(c \mid x) = \frac{P(x \mid c) P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

In addition, it is known to outperform even high-end classification methods. However, a better performance comparing to what we have tried could be achieved by defining prior probabilities as the class percentage for each author in the training set.

4.4.3. *Support Vector Machines (SVM).* Another supervised learning models that is used for classification and regression analysis. In general, an SVM classifier for an n-dimensional feature space attempts to find the (n-1) dimensional hyperplane that optimally separates the data; that is, the hyperplane which maximizes the "margin", or largest separation, between the classes. However, Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier while deleting will change the position of the hyperplane [14]. In practice, the SVM's concept of maximum margin leads to better generalization, and thus better accuracy for binary and multi-class classification problems.
For linear SVM, we can write any hyperplane in form of $w^{\rightarrow}.x^{\rightarrow} - b = 0$ where $w^{\rightarrow}$ is normal vector to the hyperplane. The parameter $\frac{b}{||w||^{\rightarrow}}$ determines the offset of the hyperplane from the origin along the normal vector $w^{\rightarrow}$.
Following constraints state that each data point must lie on the correct side of the margin.
$w^{\rightarrow}.x^{\rightarrow} + b \geq +1 if y_i = +1$
$w^{\rightarrow}.x^{\rightarrow} + b \geq -1 if y_i = -1$
We can put this together to get the optimization problem:
Minimize $||w||^{\rightarrow} subject\, to\, y_i w^{\rightarrow}.x^{\rightarrow} - b \geq 1 for i = 1, ...n$

4.5. **Model Evaluation.** Here, we define evaluation measures that we used in correlating the results.

4.5.1. *Confusion Matrix.* A confusion matrix is a table that is used to describe the performance of a classification model (or a classifier) on a set of test data for which true values is known. It is a table with 4 different combinations of predicted and actual values in form of True Positive, True Negative, False Positive, False Negative. Matrix can be represented as:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \quad ; \quad \text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False Positive}}$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad ; \quad \text{Error Rate} = \frac{\text{FP+FN}}{\text{TP+TN+FP+FN}}$$

$$\text{F1 Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall + Precision}}$$

4.5.2. *ROC Curve.* Receiver Operating Characteristics are a useful tool for evaluating and comparing performance of classifiers / predictive models. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity while false-positive rate is also known as probability of false alarm and can be calculated as: $(1 - Specifity)[15], where\, specificity = \frac{TN}{TN+FP}$ Relation between Sensitivity, Specificity, FPR and Threshold: Sensitivity and Specificity are inversely proportional to each other. So, when we increase Sensitivity, Specificity decreases and vice versa. When we decrease the threshold, we get more positive values thus it increases the sensitivity and decreasing the specificity. Similarly, when we increase the threshold, we get more negative values thus we get higher specificity and lower sensitivity. As we know FPR is (1–specificity). So, when we increase TPR, FPR also increases and vice versa [16].

4.5.3. *AUC Curve.* Area Under the Curve is the area under the ROC curve. This gives us a good understanding of how well the model performs. AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example [17]. Therefore, AUC ROC curve indicates how well the probabilities from the positive classes are separated from the negative classes.

5. Experimental setting

We are going to split the data into training, validation and testing sets. 10-fold cross validation techniques will be implemented. For prediction; we are going to experiment with multiple classification algorithms, compare the out of sample accuracy using confusion matrix and area under the curve of different models.

6. Experimental Results and Analysis

In this section, we compare our methods and models detailed in section 4 in terms of sentiment analysis, classification accuracy and better prediction models. We also analyze the impact of adding various features on the overall classification accuracy.

6.1. **Text Mining.**

We executed a robust experimentation and analysis using the dataset

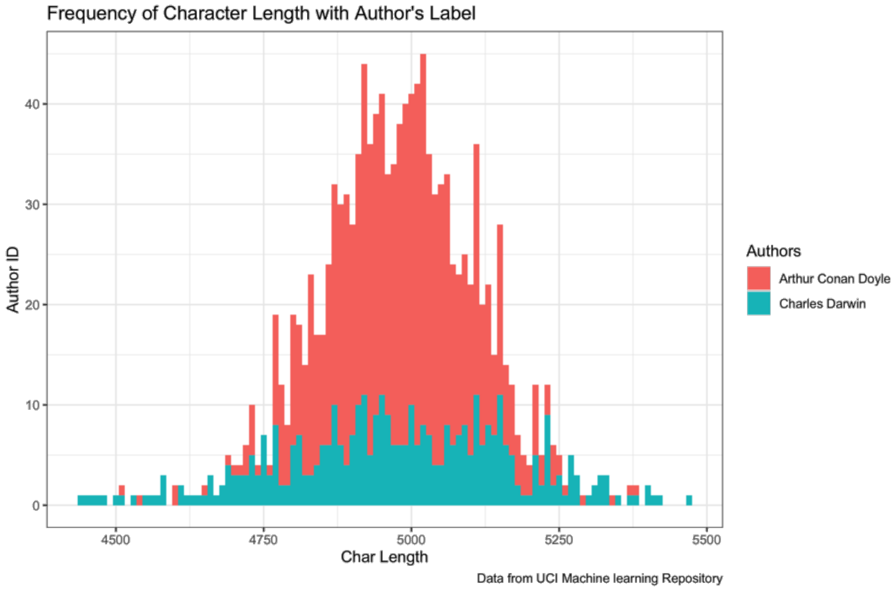


FIGURE 3. Distribution of engineered features

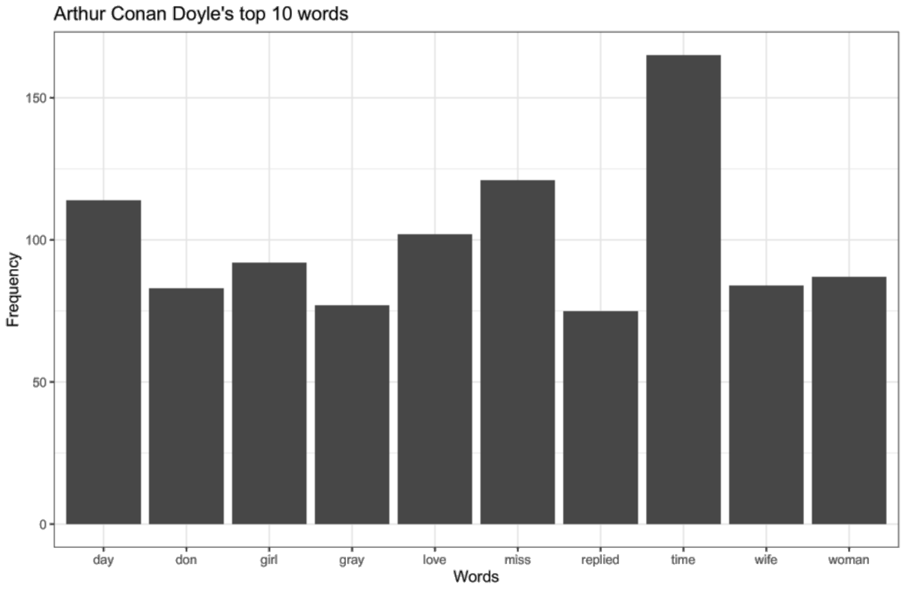


FIGURE 4. Arthur Conan Doyle's top words

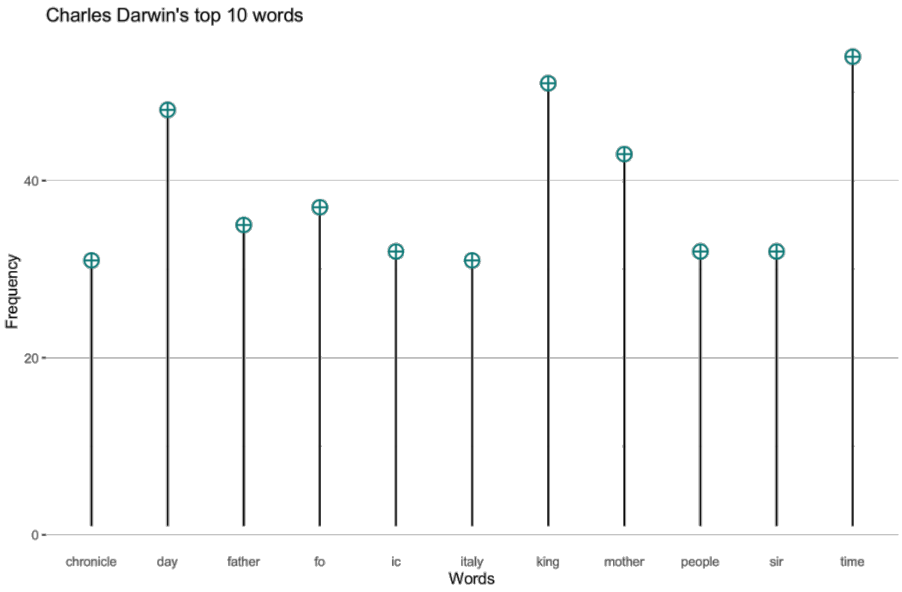


Figure 5. Charles Darwin's top words

6.1.1. *Data Preprocessing and feature engineering.* From the figures above, we can see that there exist some commonalities between the authors and their preferred choice of words.

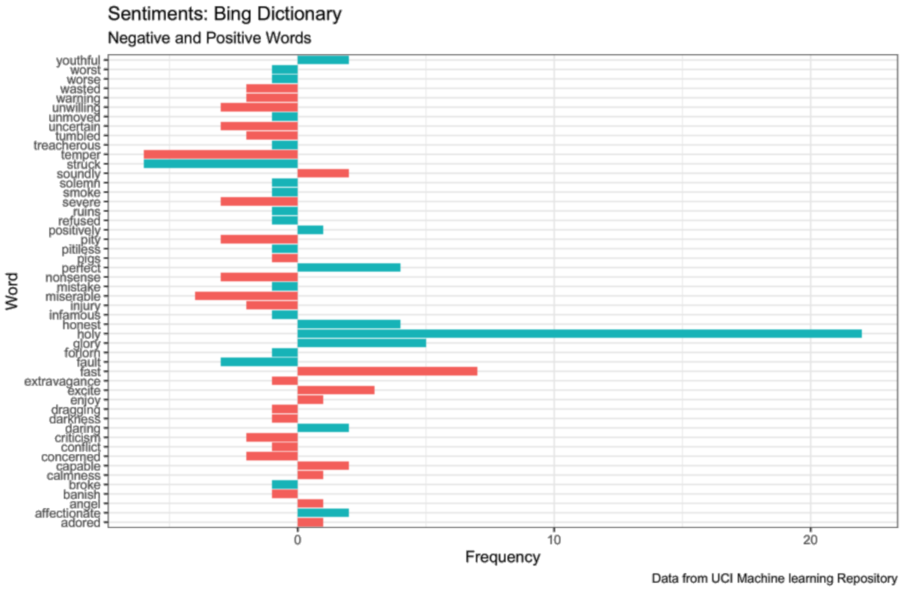


Figure 6. Top negative and positive words

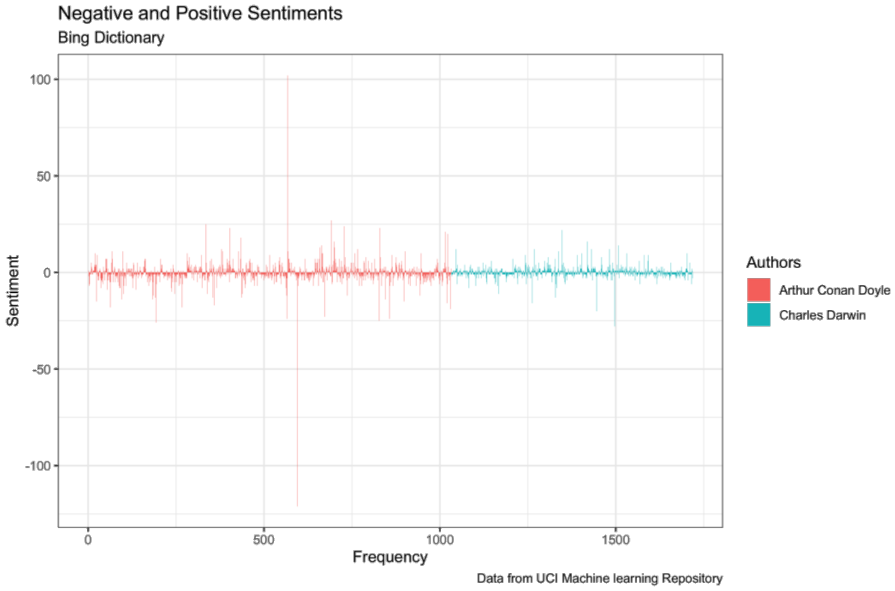


Figure 7. Negative and Positive sentiments using the Bing dictionary

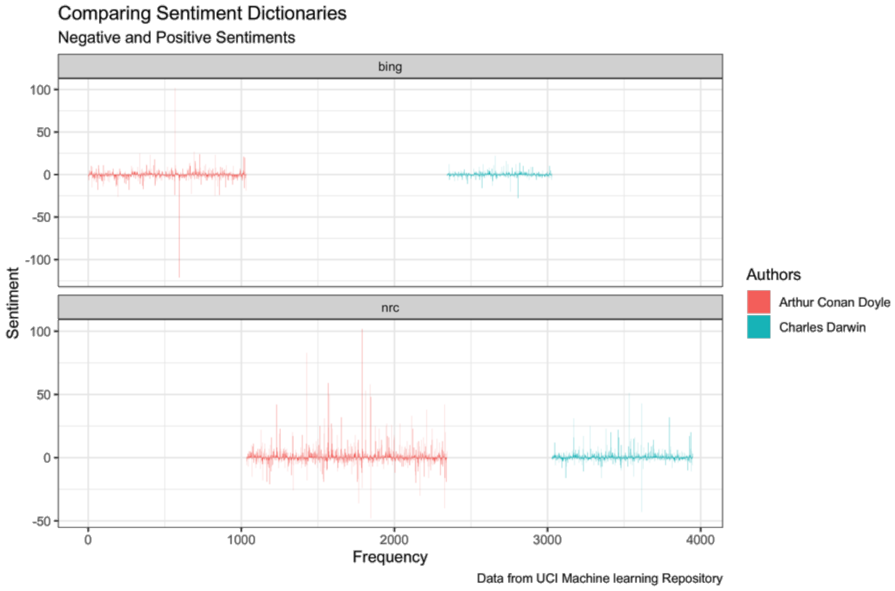


Figure 8. Comparison of the Bing and NRC sentiment dictionaries

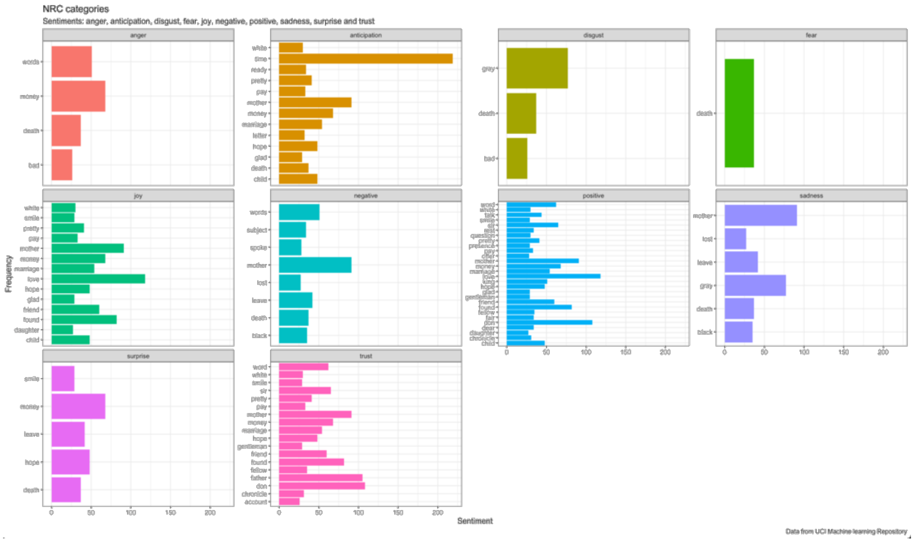


FIGURE 9. The NRC dictionary categories

6.1.2. *Sentiment Analysis.* The NRC dictionary have over ten sentiment categories, which include: anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise and trust. We immediately detect which words are categorised in more than five of aforementioned sentiment categories.

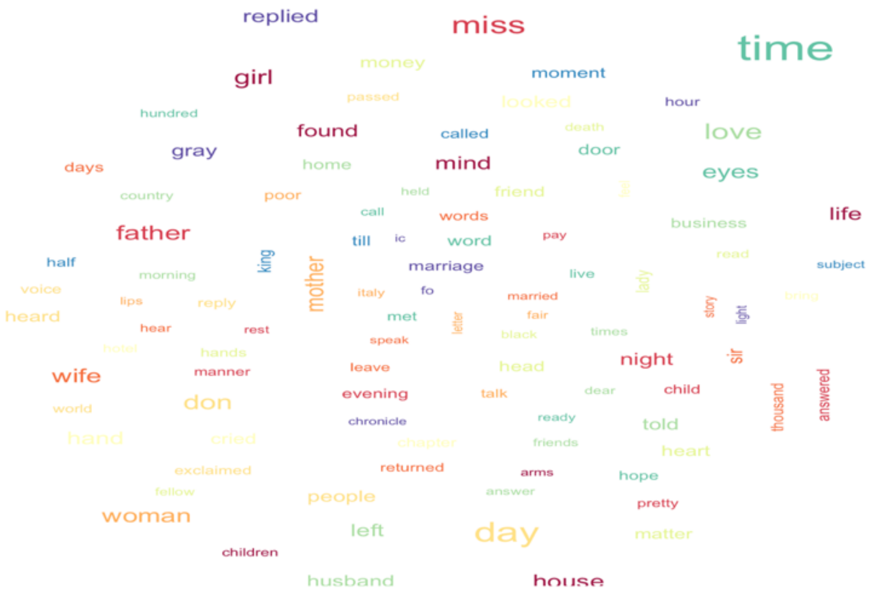


FIGURE 10. Word Cloud

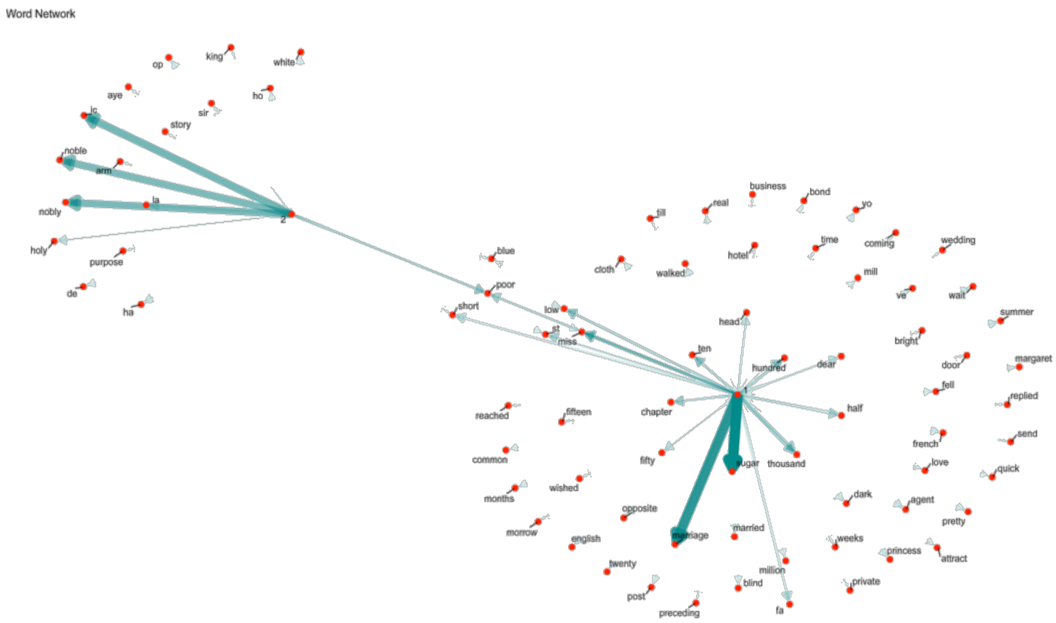


Figure 11. Word Network

The graph above shows a visualization of bigrams and their connection in the corpus. We also show display a link network to authors and the most common word they share.

6.2. **Model Evalaution.**

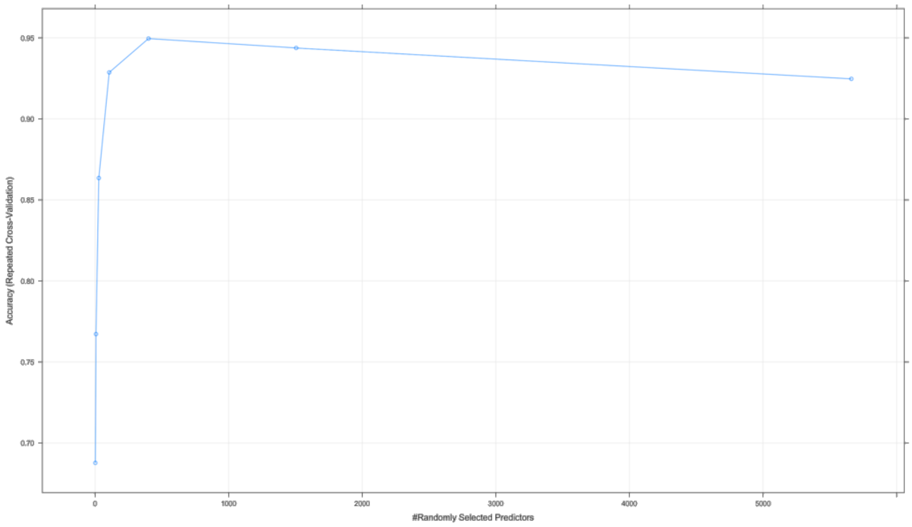


Figure 12. Cross validation on validation set

6.2.1. *10-fold cross validation.*

6.2.2. *Random Forest Classifier.*

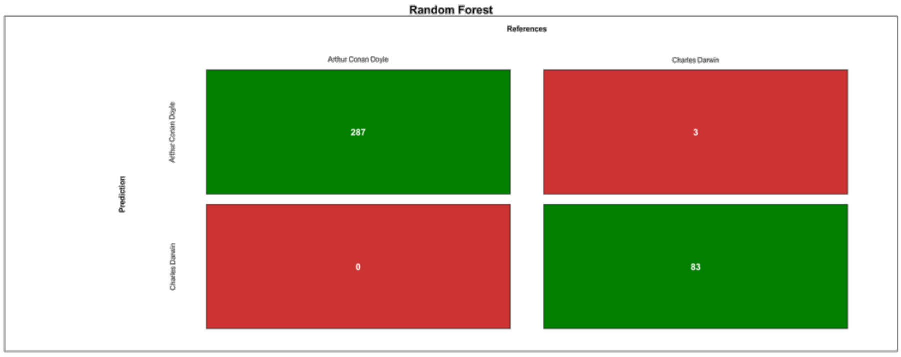


FIGURE 13. Confusion matrix

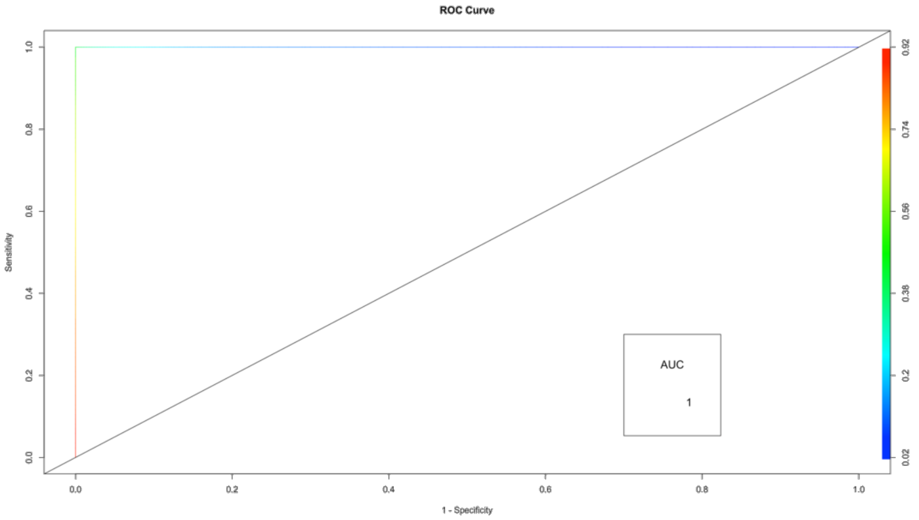


FIGURE 14. ROC and AUC curve

6.2.3. *Confusion Matrix.*

6.3. **Future Work and Conclusion.** In this report, we described our methodology for Text Mining and Authorship Prediction. We performed feature engineering and data wrangling, sentiment analysis and finally predicted authors based on a collection of text derived from their authored books. For further research, we plan to perform employ a more computational resource-intensive approach to sample more authors and train our dataset based on more stylometric features like Part of speech and sentiment analysis.

7. References

[1] Ying Zhao Justin Zobel. “Searching with Style: Authorship Attribution in Classic Literature”.

[2] Mudit Bhargava Pulkit Mehdiratta Krishna Asawa. ”Stylometric Analysis for Authorship Attribution on Twitter”.

[3] Sanderson J. and Simon G., “Short Text Authorship Attribution via Sequence Kernels, Markov Chains and Author Unmasking: An Investigation”.

[4] Đlker Nadi Bozkurt Özgür Bağlıoğlu Erkan Uyar. “Authorship Attribution”.

[5] Stamatatos E, Fakotakis N., and Kokkinakis G. “Computer- Based Authorship Attribution without lexical measures”.Computers and Hummanities, 2001 pp.193-214.

[6] Diederich Joachim, Kindermenn Jörg, Leopold Edda, and Pass Gerhard.“Authorship attribution with Support Vector Machines”. Applied Intelligence. 2003 pp.109-123.

[7] Rui Sousa Silva, Gustavo Laboreiro, Luis Sarmento, Tim Grant, Eugenio Oliveiraand Belinda Maia.Automatic Authorship Analysis of Micro-BloggingMessages ”. In International Conference on Application of Natural Language to Information Systems , LNCS 6716, pp. 161168, 2011.

[8] Anderson Rocha, Walter J. Scheirer, Christopher W. Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne R. B. Carvalho, and Efstathios Stamatatos. “Authorship Attribution for Social Media Forensics”.

[9] Dataset: https://archive.ics.uci.edu/ml/datasets/Victorian+Era+
Authorship+Attribution.

[10] K. Luyckx and W. Daelemans, “Authorship attribution and verification with many authors and limited data,” in Proc. Int. Conf. Comput. Linguistics, 2008, pp. 513–520.

[11] Wenji Mao, Fei-Yue Wang, “ New Advances in Intelligence and Security Informatics”.

[12] Alonso Palomino-Garibay1, Adolfo T. Camacho-González1, Ricardo A.Fierro-Villaneda2 Irazú Hernández-Farias3, Davide Buscaldi4, and Ivan V. Meza-Ruiz2. “A Random Forest Approach for Authorship Profiling”.

[13] Naïve Bayes. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[14] Support Vector Machinehttps://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47/

[15] ROC curve. https://acutecaretesting.org/en/articles/roc-curves-what-are-they-and-how-are-they-used/

[16] Relation. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[17] AUC https://developers.google.com/machine-learning/crash-course/
classification/roc-and-auc