Task: Classification problem

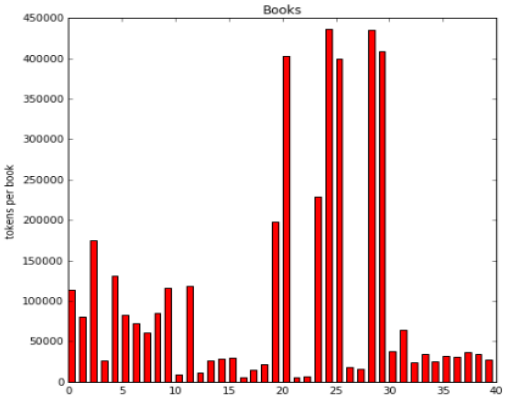
To develop a system for authorship attribution: given a number of books in textual format, a subset of which is of known author, attribute the authorship to the remaining part of books.
Programming Language: Python
Libraries used: Numpy, Scikit-Learn, nltk, glob

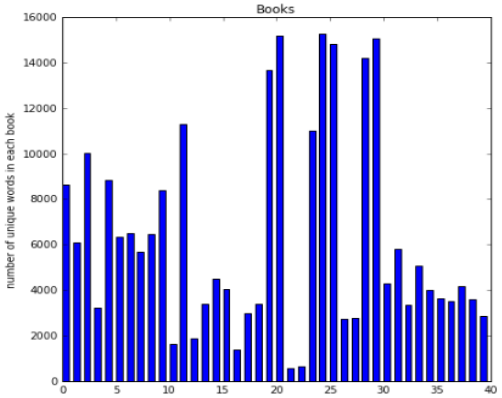
Data:

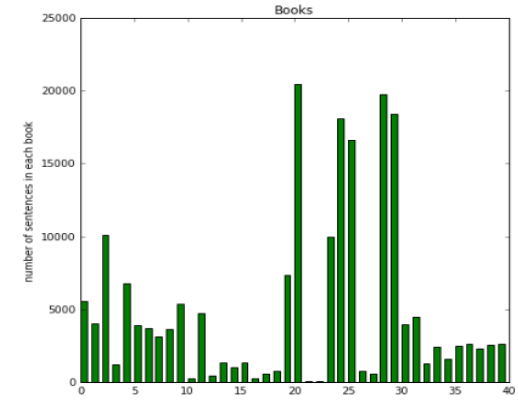
The data consists of 40 manually downloaded texts written by 4 different authors from the given websites. Hence the dataset consists of a mix of 4 different labels namely: Charles Dickens, Shakespeare, Jules Verne and Mark Twain. The dataset consists of 10 books written by each author.

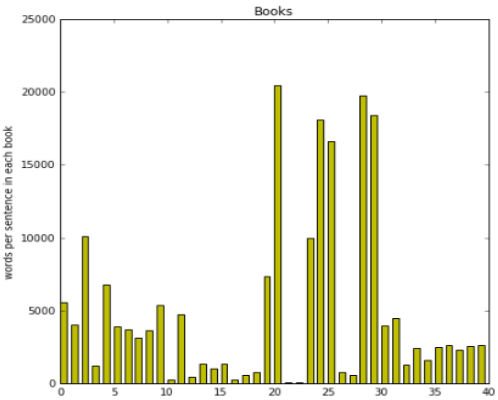
Approach:

The text from each book generates millions of tokens. The problem with so many tokens is that each token if considered as a feature will give a high variance in the data set. Thus we need to carefully select the features that are most important to determine the author of an unknown text. Hence there is a need to capture the writing style of an author through his text. The words people use and the way they structure their sentences is distinctive, and can often be used to identify the author of a particular work. For example parts of speech of the text will be a good means to determine the style of the author. The following are the plots of tokens per book (including punctuation), words per book (without punctuation), unique words per book and sentences per book.









Feature Extraction:

There are 3 main types of features that are extracted in this task.

1. **Lexical features**: average of words per sentence, sentence length variation and richness of authors vocabulary
2. **Punctuation features**: Average number of different punctuations that are frequently used by the authors.
3. **Bags of words**: Bag of words gives the frequencies of different words in each book. Thus I have taken a selected number of words that are commonly used by all authors. I have limited this number to 250 for the purpose of this assignment. However as the number of books goes on increasing, this number should be increased accordingly. CountVectorizer class of sklearn library is used to count these common words in each text of the book.
4. Could also have used parts of speech to determine the features.

These features are appended vertically to form a full feature vector matrix(X)
The labels(y) of each book have the known names of respective authors.

Classification and evaluation:

For classification, I have used 3 different classifiers from sklearn library namely Logistic Regression, Naïve Bayes and Support Vector Machines. Naïve Bayes gives an average performance. However, Logistic regression and SVM work quite well.

Sklearn's cross validation feature is used to divide the dataset into variable training and testing sets that will have different size of training and testing data for each iteration. Number of folds are set to 10 by default in the cross validation method.
For each iteration, the object of the classifier fits the given training data and tries to predict on the testing data.
For evaluation of performance, three metrics have been used namely: accuracy, precision and recall. These 3 metrics help in comparing the predicted labels to the true labels of the data. In this manner, performance can be measured for each fold in cross validation. Ultimately the means of all metrics along all folds is printed for each classifier.

Additional questions:

1. How would you assess the performances of your system?
2. Could your system be used to generate novel content such that it appears as being written by a given author?
3. Is your system scalable w.r.t. number of documents / users? If not, how would address the scalability (in terms of algorithms, infrastructure, or both)?

First 2 questions can be answer by looking at the classification and evaluation part
The answer to third question is no. We cannot say that this system is scalable. A scability issue arises when I increase the number of books. With increase in number of books, the variance of words and tokens increases and hence the number of common words that we use as bags may get affected which results in null values for some documents. Thus, I needed to increase the size of common words that needs to be used as features in order to successfully execute the program. I guess, regularization in the machine learning algorithms may help increase the scalability.

References

http://www.aicbt.com/authorship-attribution/

http://scikit-learn.org/stable/documentation.html
http://www.nltk.org/