CS 585 Natural Language Processing

Authorship attribution and finding whether two books are written by same author

Submitted To
Dr. Aron Culotta

By,
Balaji A R A20347964 balambad@hawk.iit.edu
Rajarajeswari Vaidyanathan A20362220 rvaidyan@hawk.iit.edu

Abstract:

The proposed project is to detect author of the given text by inferring characteristics which are specific to author. Additionally, we did a tool which will detect whether the two-different text were written by same author or not. The person writing style will be differed by words they choose, sentence structure, frequency of common words and sentence length. To tackle authorship attribution, we can use either supervised or unsupervised learning. We have used supervised learning for authorship prediction and unsupervised learning for finding whether two books are written by same author or not. This paper will also deal in finding the similarity between the authors based on their linguistic style each authors exhibit. Authorship detection is a kind of classification problem. For example, some authors prefer to write long sentences without any punctuations, whereas others prefer to write a concise gist. In our project, we have extracted different features from the texts and the fit model based on them, thereby calculating the confusion matrix which gives us various measures like error-rate, accuracy, precision and recall. Our solution assists in many other areas like plagiarism detection, email filtering, and to check the originality of the content.

At the end of our project we identified that each author can be uniquely identified using lexical and punctuation features, syntactic features, bag of words features. it's difficult to find exact set of features which are helpful to clearly identify authors of the given text to clearly cluster the two texts. Additionally, we found that it is difficult to select good features for unsupervised learning. During our testing, we find that the results are not stable while clustering. For example, if we make a minor change to our feature selection (e.g. normalization method), or even run k-means again the clusters would change. This clearly tells us that clusters are not well separated in feature space.

Introduction:

Authorship attribution is the process of determining the writer of a document. There are lots of classification techniques available to solve this problem. In our project, we have used support vector machines(svm) in authorship attribution and unsupervised (clustering) classification techniques in detecting whether two texts are written by same author or not. We performed various experiments with articles gathered from http://www.gutenberg.org. We performed experiments on different features extracted from these texts using svm, and combined those

results to improve our success rates. We identified that svm gives satisfactory results. According to experiments, the success rates dramatically changes with different combinations, however the best among them are svm classifier with bag of words features, syntactic, lexical and punctuation features. Our solution assists in many other areas like plagiarism detection, email filtering, and to check the originality of the content.

Related Work:

There are hundreds of researches conducted about this subject in the recent years. With the increasing amount of data available 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. Stylometry - the statistical analysis of literary style - complements traditional literary scholarship since it offers a means of capturing the often-elusive character of an author's style by quantifying some of its features [1]. Most stylometric studies employ items of language and most of these items are lexically based.

The usefulness of function words in Authorship attribution is examined by Argamon and Levitan [2]. The authors conducted experiments with support vector machine classifiers in twenty novels and they obtained success rates above 90%. They concluded that, using function words is a valid and good approach in authorship attribution. According to last researches in 2001, Stamatatos, Fakotakis, Kokkinakis [3] 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. In 2003, Joachim Diederich and his collaborators conducted experiments with support vector classifiers and detected author with %60-80 success rates with different parameters [4]. Kjell [5] performed experiments with neural networks and Bayesian classifiers in this area and obtained about 80-90% success.
Most of the researchers aimed to consider both stylistic and topic features of texts. In this classification problem.

In our project, we have combined bag of words, lexical and punctuation, syntactic features in SVM for the authorship attribution which provided us with 75% accuracy on the testing set.

Approach for authorship attribution:

A complete authorship attribution process consists of collecting texts from http://www.gutenberg.org and preprocessing text to remove metadata information about the text. Processed data are the observations to be classified in some sense; a feature extraction mechanism that computes numerical or symbolic information from the observations; and a classification or model that does the classification observations which rely on the extracted features from the preprocessed texts.

First, we developed a crawler to download from http://www.gutenberg.org, once the text is downloaded, we process the texts to remove metadata information about the text. Most of the texts are written by following authors alexander dumas Herman Melville, jane austen and leo tolstoy. Second, we extracted following features from the text to help in classifying. Following table shows the extracted features and they help us in identifying characteristic.

Number of hapax legomena divided by number of unique words	Hapax legomena
Number of dis legomena divided by number of unique words	Dis legomena
Number of unique words divided by number of total words	Richness
Flesch readability score divided by 100	Readability
No. of sentences of length in the range [1, RANGE] divided by the number of total sentences	Sentence length distribution
No. of words of length in the range [1, RANGE] divided by the number of total words	Word length distribution
No. of nominative pronouns per sentence in the range [1, RANGE] divided by the number of total sentences.	Pronouns distribution
No. of (coordinating + subordinating) conjunctions per sentence in the range [1, RANGE] divided by the number of total sentences.	Conjunction distribution

Richness and readability features help us to identify author's vocabulary and it is used as a discriminating feature. Additionally, the function words pronoun, conjunction distribution also acts as a discriminating feature. We have used LinearSVC from the sklearn library to classify collected texts using the extracted features.

Results: Authorship prediction using SVM

Below image shows the results using all the features listed in the previous section.

```
Author classification started...
Processed 31 books from 4 authors with 1620817 total words in 43.294s

31 samples in 4 classes

Accuracy on training set: 1.000

Accuracy on testing set: 0.769

Confusion Matrix:
                 alexandre-dumas  herman-melville  jane-austen  leo-tolstoy
alexandre-dumas                2                0            0            1
herman-melville                0                1            0            1
jane-austen                    0                0            4            0
leo-tolstoy                    1                0            0            3

Result:
           alexandre-dumas  herman-melville  jane-austen  leo-tolstoy
precision         0.666667         1.000000          1.0     0.600000
recall            0.666667         0.500000          1.0     0.750000
f1                0.666667         0.666667          1.0     0.666667

Average f1s:
0.75
```

we ran our classification algorithm on various combinations of features and below table represent the result of our experimentation.

Sentence length distribution Word length distribution Pronoun distribution Conjunction distribution	52%
Hapax legomena Dis legomena	35%

Richness Readibility	
Hapax legomena Dis legomena Sentence length distribution Word length distribution	61.4%
Sentence length distribution Word length distribution Pronoun distribution	84%
All features	75%

Results: Check Authors are same or different for given two texts

To check authors are same or different for given two texts we used KMeans cluster algorithm from sklearn library. We used the same data collected for this problem as well. To solve this problem, we collected following three type of features.

Lexical and punctuation features

- Lexical features:
  - The average number of words per sentence
  - Sentence length variation
  - Lexical diversity, which is a measure of the richness of the author's vocabulary
- Punctuation features:
  - Average number of commas, semicolons and colons per sentence

Bag of Words features

- Our second feature set is Bag of Words, which represents the frequencies of different words in each text. This feature vector is commonly used for text classification.

Syntactic features

- For our final feature set, we extract syntactic features of the text. Part of speech (POS) is a classification of each token into a lexical category (e.g. noun). NLTK has a function for POS labeling, and our feature vector is comprised of frequencies for the most common POS tags

To perform tests, we created pair of books, then extracted three types of features and performed KMeans clustering to see whether both are classified into same cluster or not. Below are the results for our approach.

Evaluation results:
Number of file pairs classified correctly: 25
Number of file pairs Incorrectly classified: 10
Accurate percentage: 71.43%

Steps to test our code:

[1] First install below 3 Models from nltk by running nltk.download() from python terminal
- Averaged Perceptron Tagger
- Treebank Part of Speech Tagger
- Punk Tokenizer Models

[2] To test authorship prediction run `python svm_author_pred.py` from author-prediction folder from the cloned directory.
[3] To check authors are same or different for given two texts run `python author-classify.py` from author-attribution folder from the cloned directory.

Conclusion and future work:

In this project on authorship attribution, we used different feature sets with our data set, which are bag of words, lexical and punctuation, syntactic features and performed experiments on these feature sets using SVM. One possible solution to improve our results using more complex prediction models such as recurrent neural networks which are well suited for the authorship attribution.

References:

[1] Laan, N.M. "Stylometry and Method. The Case of Euripides", Literary and Linguistic Computing, 10, 271-278, (1995).
[2] Shlomo Argamon, Levitan Shlomo. Measuring the usefullness of function words for Authorship Attribution. Proceedings of ACH/ALLC Conference 2005 in Victoria, BC, Canada, June 2000.
[3] Stamatatos E, Fakotakis N, and Kokkinakis G. “Computer- Based Authorship Attribution without lexical measures”. Computers and Hummanities, 2001 pp.193-214.

[4] Diederich Joachim, Kindermenn Jörg, Leopold Edda, and Pass Gerhard. “Authorship attribution with Support Vector Machines” . Applied Intelligence. 2003 pp.109-123
[5] Bradley Kjell. Authorship Attribution of Text Samples using Neural Networks and Bayesian Classifiers.
[6] Ying Zhao Justin Zobel. “Searching with Style: Authorship Attribution in Classic Literature”.
[7] http://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a106.pdf
[8] http://nifty.stanford.edu/2013/craig-authorship-detection/
[9] http://www.aicbt.com/authorship-attribution/