# Authorship Attribution at the Paragraph Level using SVMs

## Introduction

Using statistical analysis to determine authorship has proven helpful in many spheres. Mosteller and Wallace conducted one of the most important early studies on author attribution, using a Naive Bayes classifier to discover the author of *The Federalist Papers*.[1] But in addition to its use for literary research, techniques for authorship attribution have been used for gathering intelligence about terrorists, identifying authors of criminal messages and settling copyright disputes.[2] In this project, we are interested in using support vector machine (SVM) classifiers in order see if we can determine how much different coauthors contribute to a given paper.

## Related Work

As mentioned, Mosteller and Wallace used the counts of high frequency language filler words (*an, to, upon...*) as features to, for the first time, determine the authorship of 12 of the anonymously written *Federalist Papers* with Naive Bayes. An even earlier method by Mendenhall used sentence and word length counts as features.[2]

A similar method is the multi-variate Bernoulli model, which only stores a binary value for whether or not a word is present in a document, discarding the word count. It has been shown to perform better on documents with smaller vocabulary sizes, but has impaired performance at larger vocabulary sizes.[3]

The Chain Augmented Naive Bayes classifier (CAN) is a variant that additionally uses $n$-gram information to improve performance, with accuracies of up to 96% when more sophisticated smoothing techniques are used.[4] The SVM is a more general purpose classifier that has also been applied to text classification, with author attribution accuracies of 60-80% on newspaper data and 70-95% on online message data.[2] Diedrich, et. al. have further shown an SVM is capable of virtually perfect precision when attributing authorship using various features.[5] We will compare our performance to that of Diedrich's paper.

## Approach

We implemented a support vector classifier in an attempt to identify the authors of small portions of text within a document written by multiple authors. We used, as our documents, papers written by a local computer scientist, whom we will refer to as author $A$, as well as papers written by his students and coauthors, referred to as authors $B$, $C$ and $D$. Our ultimate goal was to ultimately see how confidently we could look at text down to the paragraph level in papers with multiple authors, and distinguish who was most likely to have written different portions of the paper. To train and test the classifier, we first looked at papers independently written by each of the authors.

We used features based on the results from Diederich, et. al., in which they first classified text using relative word frequencies as features, and then using features referred to as *tag-words*. While those authors obtained better results with word frequencies, they had a much larger dataset of thousands of documents, which varied over different topics. Since our training papers covered roughly the same topic, and because topical features would not give any discriminating information between the authors of co-authored papers, we opted for the tag-word features, which are considered to be more independent of context.

We obtained tag-words by concatenating a word stem and the part of speech tag for a given word, but only including the part of speech tag for adjectives, nouns, verbs and numbers. So for a phrase like *blue dogs run*, the corresponding tagwords, were, respectively, *JJ, NNS VBG* (which represent adjective, plural noun and gerund verb). For phrases with words of different parts of speech, such as *the honestly*, the stemmed words would be *the, honestli*, and the POS tags would be *DT, RB* (for determiner and adverb), resulting in tagwords 'the-DT' and 'honestli-RB'. For each paragraph, we then constructed vectors with the frequencies of each of these tagwords, and used SVMs to classify the authors. For word stems, we tested and compared both the porter stemmer and WordNet lemmatizer, and used a POS tagger based on the Penn Treebank project, all from the Natural Language Toolkit (NLTK) python module [6]. We used a linear SVM from the scikit-learn python library.

The hope in using tagwords was that the relative frequency of the parts of speech that different authors used would contain more information than the raw word counts that Naive Bayes takes into account. Furthermore, if documents in a sample are all discussing the same topic, say *wearable computing*, it is unlikely that the words *wearable* and *computing* would be able to tell a lot about the author. Thus, we did not use the stemmed words of adjectives, nouns and verbs to construct our tagwords, in order for the classifier to be more content-independent, while we did use the stems of words like *the* and *very*, in order to hopefully glean more information about the author's writing style.

After getting the tagwords in all the paragraphs written by a given author, we used the tagwords to create feature vectors to input into the SVM. Diederich, et. al. describe different helpful vector transformations for SVMs,

such as raw frequency (raw tagword counts) relative frequency (tagword frequency divided by number of different tagword types), logarithmic relative frequency (log of relative frequency, denoted in this paper as 'LogRel'), and used normalizations such as the $L_1$ and euclidean norms. For each author, we tried different combinations of the above to determine the vector transformation with the most predictive power for each author.

## Results

We performed 3 major tasks:

1. Testing the classifier on paragraphs from documents written by a *single known author*, using documents from *all* of the authors as training data.

2. Testing the classifier on paragraphs from documents written by a *single known coauthor*, using *documents solely written by that coauthor and documents solely by author A* as training data.

3. Testing the classifier on paragraphs from documents written by *2 authors*, using documents written by those two authors individually.

For the first two tasks, we knew the authors of the paragraphs, and thus got more reliable results. For the third task, the author of the paragraphs was unknown, and our corresponding results should thus be considered unreliable at best.

For the first step, we evaluated the performance based on the precision, recall and F-score averages after performing 10-fold cross validation. The results can be seen in Table 1. These are the results when the classifier was trained to distinguish one author out of all 4 different authors, and we did this to compare our results with those by Dietrich, et. al. It shows that the WordNet lemmatizer and euclidean ($L_2$) norm worked best for all of the authors, and that using the raw frequency word count as features worked best most of the time. The maximum F-score was 0.88.

For our application, however, we already know that a given paragraph in a co-authored paper was written by one of 2 authors, and so we can rule out all of the other authors for our classifier. Table 2 thus shows the 10-fold cross-validation results for documents written by the students, in which we only trained on the student and author $A$. The lemmatizer and $L_2$ norm were used in all cases, and the LogRel normalization seemed to give the best results. The highest F-score for these was 0.94.

Finally, we classified coauthored papers with training samples from the individual authors. Ideally, we would like a classifier that could give probability estimates for how much a paragraph looks to be influenced by a specific author, but this was not possible since SVMs only give binary classification predictions. We decided instead to run the classifier 100 times for each co-authored docu-

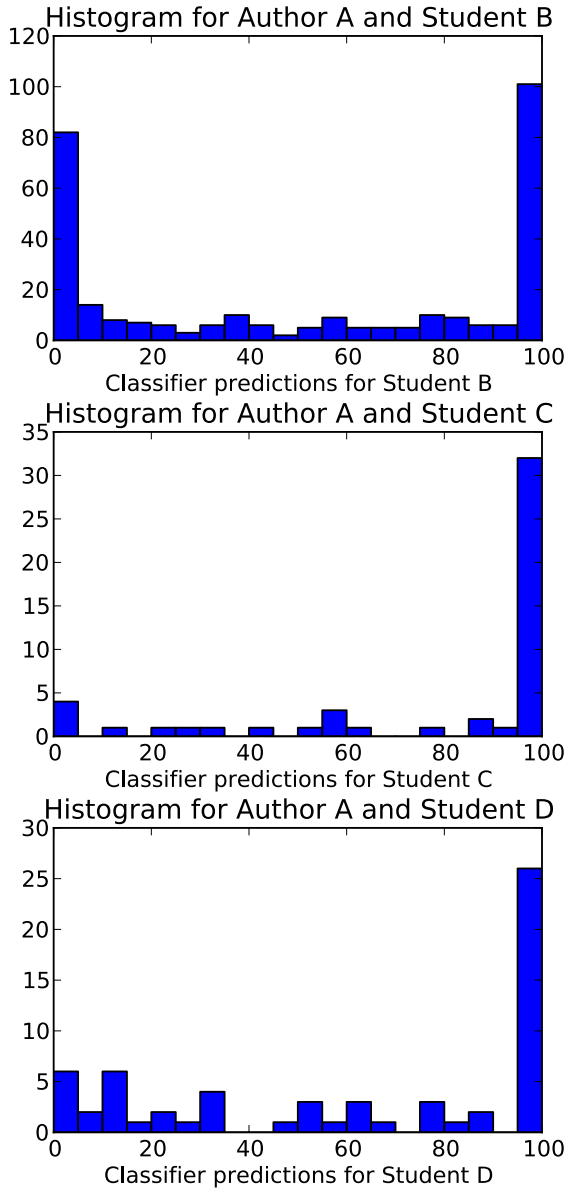


Figure 1: Histograms of number of paragraphs classified as being written by Author $A$ or a student coauthor at different frequencies. Bins to the far right represent the number of paragraphs that the classifier attributed to the student about 100% of the time. Bins near the middle represent the number of paragraphs that were attributed to each author about an equal number of times.

ment, using only half of the training examples (randomly chosen) each time. A histogram of the results can be seen in Fig. 1.

For each histogram, the bin on the far left shows how many paragraphs the classifier predicted to be written by author $A$ for approximately all of the trials, and the bin on the far right shows how many were predicted to be written by the student for almost all of the trials, based on the random samples of training data. The papers coauthored by authors $A$ and $B$ appear to have been divided most equally, while the number of paragraphs that the classifier ascribed to author $A$ looks to be around the noise threshold for the paper coauthored with author $C$.

## Discussion

As seen in the tables, most of our results were much less accurate than those of Diedrich, et. al, whose typical F-score was about 0.9. We believe this is largely because of their higher sample size. At the paragraph level, our data was much smaller, and we had a limited number of available papers written by the given authors. In contrast, their group used a corpus of more than 2600 documents, with dozens of papers for some authors. There was, however, a noticeable improvement when we only try discriminating between two authors, rather than between a given author and all the rest. This improved the accuracy of the classifier to around the accuracy of Diedrich's results.

Our small amount of training data, however, makes us less confident in the results of attributing authors at the paragraph level. Even beyond the fact of such small sample sizes, our model assumes that papers are discretely written by different individuals at the paragraph level, when the writing process leaves much room for authors changing the tones and expressions they use while writing about a given topic, as a result of the research environment.

In general, this task is especially difficult given the lack of training data, as coauthors do not typically annotate their papers by labeling the author of each individual paragraph. When discriminating against 2 authors, however, the SVM classifier performed better, and improved even further for the author with the most available training data, author $C$. It would thus be worthwhile to investigate authorship attribution among coauthors with a larger set of data, and possibly with more classifiers.

# References


[1] F. Mosteller and D. Wallace. *Inference In An Authorship Problem.* Journal of the American Statistical Association, Volume 58, Issue 302 (1963), 275-309.

[2] E. Stamatatos. *A Survey of Modern Authorship Attribution Methods.* University of the Aegean, Greece.

[3] A. McCallum, and K. Nigam. *A Comparison of Event Models for Naive Bayes Text Classification.*

[4] F. Peng and D. Schuurmans. *Combining Naive Bayes and n-Gram Language Models for Text Classification.* ECIR (2003), 335-350.

[5] J. Diederich, J. Kinderman, E. Leopold and G. Paass. *Authorship Attribution with Support Vector Machine.* Applied Intelligence 19 (2003), 109-123.

[6] S. Bird. *NLTK: The Natural Language Toolkit.* University of Melbourne, Australia.


Table 1: Results of the SVM on individually-written papers, averaged after 10-fold cross-validation, along with the feature frequency transformations that yielded the best results. The training data consisted of individually-written papers from *all* of the authors.

| | Fscore | Precision | Recall | Stemming | Normalization | Num. Paragraphs | Ave. Words/ paragraph | Total Words (×1000 ) |
|---|---|---|---|---|---|---|---|---|
| Author $A$ | 0.57 | 0.72 | 0.48 | Lemma. | L2, LogRel | 205 | 97 | 20 |
| Author $B$ | 0.64 | 0.63 | 0.65 | Lemma. | L2, Raw freq. | 241 | 95 | 23 |
| Author $C$ | 0.88 | 0.85 | 0.90 | Lemma. | L2, Raw freq. | 1010 | 56 | 57 |
| Author $D$ | 0.61 | 0.66 | 0.57 | Lemma. | L2, Raw freq. | 318 | 101 | 32 |

Table 2: Results of the SVM on individually-written papers. The training data consisted of papers of only the given coauthor and author $A$.

| | Fscore | Precision | Recall | $A$ Fscore | $A$ Precision | $A$ Recall | Normalization |
|---|---|---|---|---|---|---|---|
| Author $B$ | 0.85 | 0.86 | 0.85 | 0.86 | 0.89 | 0.84 | L2, LogRel |
| Author $C$ | 0.94 | 0.90 | 0.98 | 0.63 | 0.83 | 0.52 | L2, Raw Frequency |
| Author $D$ | 0.85 | 0.84 | 0.85 | 0.71 | 0.76 | 0.67 | L2, LogRel |