Authorship Attribution of Online Reviews

Aparna Budhavarapu, Aashrit Mathur

Department of Computer Science
University of Houston
Houston, TX, 77004
abudhavarapu@uh.edu | amathur3@uh.edu

Abstract. Given a set of authors, authorship attribution is the task of figuring out who, if any of them is the actual author of a piece of text [1]. Traditional authorship attribution techniques do not tend to scale well with increasing number of authors as we typically witness reviews online. They are designed to work on a small number of authors. We have devised a method of performing authorship attribution on a large set of possible candidate authors. Our results show that by combining multiple authorship verification models, we are able to successfully associate the reviews to their rightful authors about 89% of the time. We propose that our approach can be used instead of employing multi-class classifier model, which can tend to take more time for training and may obtain results worse than our method.

1. Introduction

With the advent of social media, massive amounts of social media data is being recorded all around the world. Much of the research in the past has been done on traditional authorship attribution methods that deal with a very small set of authors. They do not tend to word well with increasing amounts of data, the kind of data that is being generated online. In order to keep up with the enormous increase in this data, we need a scalable authorship attribution method that can work for organizations like Amazon and Walmart.

Authorship Attribution is a method that has its advent in 19th century. It is a method used to identify the author of a questioned document, given a set of known documents written by various authors. Traditionally, it was used on long texts like books or novels. However, employing this method to scale for the huge amounts of data being generated online is relatively new. We have proposed a model that can identify the author of a particular online review. We can use our system in current times, as there are many online e-commerce websites that have numerous reviews for each of the products that they sell. We have

witnessed many cases where people have created multiple accounts and have been providing the consumers with fake reviews. There are cases where people want to promote the sales of their product by providing outstanding reviews. There have also been cases where negative reviews about a competitor's product have been made to hinder their sales. We can use authorship attribution to identify the individuals that create multiple accounts by observing the writing pattern of the reviews.

The various challenges that authorship attribution faces when being used for online reviews include:

- Reviews are generally shorter texts than books or blogs
- They are not focused on a certain set of topics
- There are potentially many number of authors who might have written the review
- Identifying fake reviews is tough as authors generally change their writing style to avoid getting caught

Our model tries to employ authorship attribution for online product reviews by tackling the aforementioned challenges. We have devised a verification technique through which we will identify the authorship of the reviews. The dataset that we used for this paper consists of over 2,573,341 amazon product reviews that have been written by 100,001 authors. Our contribution includes loading and working on the review dataset given to us and employing methods to associate the unknown documents to their authors as effectively as possible.

2. Review Dataset

To perform the task of authorship attribution, we have a dataset containing reviews written by various authors for multiple products available on Amazon [2]. Apart from the text, there are various other attributes that are included in the dataset, like reviewer ID, date the review was posted and star ratings that can be leveraged for the purpose of authorship attribution. These reviews were posted during the duration between June 1996 and October 2012.

The dataset consists over 2,573,341 product reviews written by 100,001 authors. Apart from this, each review has around 240 tokens per review on an average. The review count per author varies significantly throughout the dataset. For some set of authors, it would be difficult to perform authorship attribution, as there is not enough text to identify and analyze their writing styles. Therefore, we opted to leave such authors out from our final dataset and chose authors that had written at least 50 reviews.

3. Implementation Method

As we have learnt by now, we are dealing with a large set of authors. It would be difficult to train a multi-class, combined classifier model. Therefore, we have decided to approach this problem by breaking it down into multiple authorship verification problems. Here, we train the verifiers on the documents (D_i) for each of the authors (a_i) separately as multiple verifier models. Each such verifier will identify if the author (a_i) it is associated with has written a given review in question or not.

Each verifier model is modeled in the following way: For each author, we define a positive set and a negative set. Positive set includes the reviews actually written by that author and negative set includes the reviews written by all other authors. We know that since our dataset consists of far more negative reviews compared to positive reviews for a particular author, we have limited both the negative set and positive set to 50 reviews each.

We followed a method, where there might be an overlap between the authors whose reviews are present in the training set and the authors whose reviews are present in the test set.

For our implementations, we took 80% of the data for training the model and the remaining 20% for testing. We performed 5-fold cross-validation to obtain reasonable metrics for accuracy, precision, recall and F1-score.

4. Features

Before we started extracting our features, we performed simple preprocessing on the data by removing URLs and converting the text to lowercase for most of our features. We have used features that have been used earlier for large-scale authorship attribution on online reviews [1].

Lexical: These feature sets include character unigrams, bigrams and trigrams along with word unigrams. These help to identify the writing styles of the authors that show affinity to certain words more than other authors.

Syntactic: We extract part-of-speech (POS) tags of words used by the authors to understand the sequential writing style of authors using POS unigrams, bigrams and trigrams.

Writing Density: This feature can also assist to distinguish authors' styles. For our model, we have used average number of characters per

word, average number of syllables per word and average number of words per sentence.

Readability: Readability features are used to assess the complexity of a piece of text written by an author. For our model, we have used the Flesch-Kincaid grade level, Gunning Fog index, Flesch reading ease, Smog index, Automated readability index, Coleman-Liau index, Linsear Write formula, Dale-Chall readability score and Difficult words [3].

Part-of-Speech (POS) Trigram Diversity: This is another feature that helps identify writing styles. It is the number of unique POS trigrams encountered by the total number of POS trigrams found in the review.

Stopword Frequency: This stylistic feature can be measured as the number of stopwords in a text divided by the total number of words in that text.

Average Word Frequency Class: This feature helps identify how frequently an author is likely to use unique words that are generally not used in a language [4]. The frequency of unique words found in a language are generated in the range of '0' to '19', with '19' signifying the most uncommon words used. We took 2 datasets, PAN-12 [5] and PAN-13 [6], and 10 novels [7] from the English language to generate our corpus. We took the frequency class of all the words present in the text and normalized it by the total number of words found in the text.

Punctuation: This is another simple stylistic feature measured as the number of punctuations found in a given text normalized by the total number of words found in the text.

Word Capitalization: This feature is a simple, yet effective technique of understanding the writing style of an author. It is measured as the number of words starting with a capital letter divided by the total number of words found in the text.

5. Results and Observations

We performed two experiments on 1000 authors for 50 reviews per author from the Amazon reviews dataset. In the first experiment, we had taken the first 50 reviews encountered in the dataset, written by the author, as positive set and the first 50 reviews encountered in the dataset, not written by that author, as the negative set.

We ran this experiment by using multiple classifier models like Decision Tree, Naïve Bayes, Neural MLP, Logistic Regression, Support Vector Machine (SVM) and performing Ada Boosting on 30 decision trees

in sequence. Figure 1 shows the results of accuracy, precision, recall and f1-score for each of the mentioned classifier.

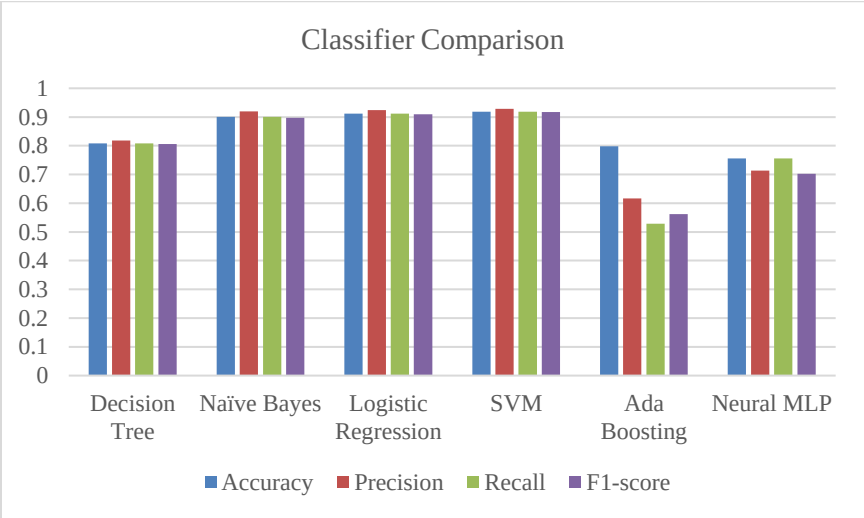

Figure 1: Comparison between Decision Tree, Naïve Bayes, Logistic Regression, Support Vector Machine, Ada Boosting and Neural MLP classifiers

From the above results, we observed that SVM provides the best performance among the above classifiers followed by Logistic Regression. Ada Boosting for 30 decision trees in sequence showed poor results suggesting that the decision tree classifier was stable and did not require performing boosting.

Table 1 shows the results of training the SVM model on 200, 400, 600, 800 and 1000 authors, 50 reviews per author, performing 5-fold cross-validation. This is followed by its illustration in Figure 2.

Performance Metrics	Number of Authors				
	200	400	600	800	1000
Accuracy	0.9418	0.9374	0.9297	0.9249	0.9188
Precision	0.9489	0.9452	0.9381	0.9338	0.9284
Recall	0.9418	0.9374	0.9296	0.9249	0.9188
F1-score	0.9411	0.9366	0.9287	0.9238	0.9177

Table 2: Performance metrics comparison by training the SVM model using 5-fold cross validation for 50 reviews per author

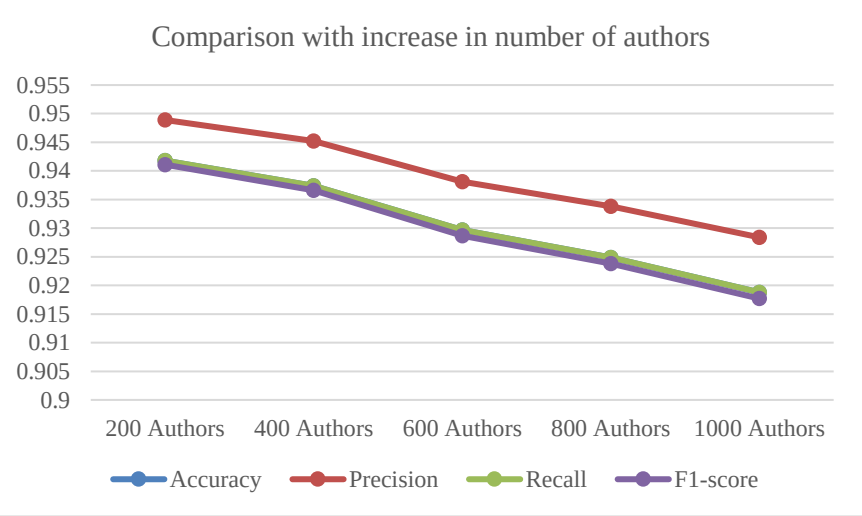

Figure 2: Illustration of the comparison by training the SVM model with increasing number of authors

From the above results, we observed that our model's performance does not completely fall with increasing number of authors. This leads us to suggest that this model can be scaled to even more than 1000 authors and retain reasonably good performance.

Table 2 shows the comparison between the performance metrics by training the SVM model on 1000 authors using 30, 50, 70 and 90 reviews per author. This is followed by its illustration shown by Figure 3.

Performance Metrics	Number of reviews per author			
	30 reviews	50 reviews	70 reviews	90 reviews
Accuracy	0.8994	0.9188	0.9304	0.9259
Precision	0.916	0.9284	0.9367	0.9314
Recall	0.8994	0.9188	0.9304	0.9259
F1-score	0.8965	0.9177	0.9298	0.9254

Table 2: Performance metrics comparison by training the SVM model using 5-fold cross validation for 1000 authors

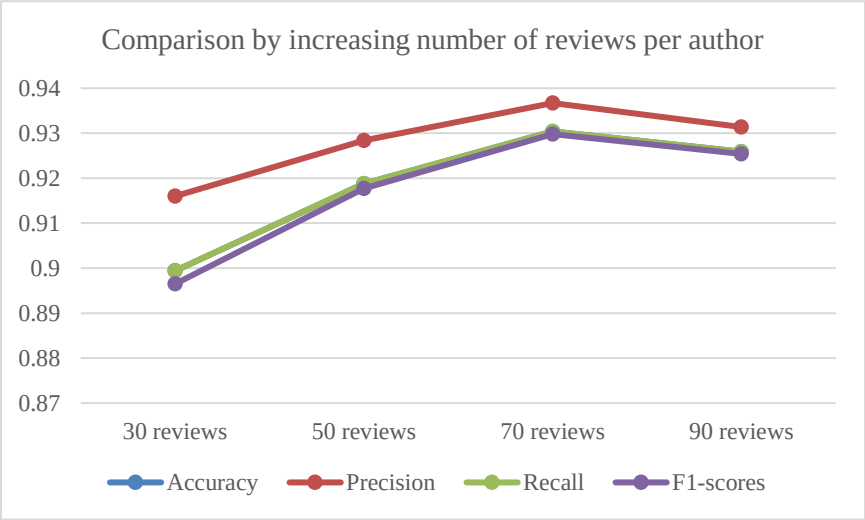

Figure 3: Illustration of the comparison by training the SVM model with increasing number of reviews per author

The above results suggest that the model works well with increasing number of reviews per author.

In the second experiment, we divided the first 1000 authors, ordered in decreasing order of number of reviews per author, in the dataset into two clusters of 500 authors each. If the author in question is from cluster A, then its negative set is generated using the reviews written by authors in cluster B. Table 3 shows the metrics comparing both the experiments for 1000 authors for 50 reviews per author using SVM and performing 5-fold cross-validation.

Experiment	Accuracy	Precision	Recall	F1-score
Method 1	0.9188	0.9284	0.9188	0.9177
Method 2	0.8966	0.9055	0.8966	0.8956

Table 3: Performance metrics comparison between the two approaches

6. Implementation Issues

We faced many issues while trying to load the data. It took us more than 24 hours to load the entire Amazon reviews dataset into our local system.

We wanted to implement generalized feature selection to obtain top 5000 features. Since we built an n-verifier model, where each verifier is associated to its author, it was difficult to obtain generalized feature importance. Each verifier has a different feature importance graph. We tried to visualize the feature importance for additional features that we

added. The following figures show the results of performing feature importance on three different authors.

0. Punctuation count
1. Flesch reading ease
2. Difficult words
3. Smog index
4. Automated readability index
5. Coleman-Liau index
6. Linsear write formula
7. Dale-Chall readability score
8. Average word frequency class
9. Capital count

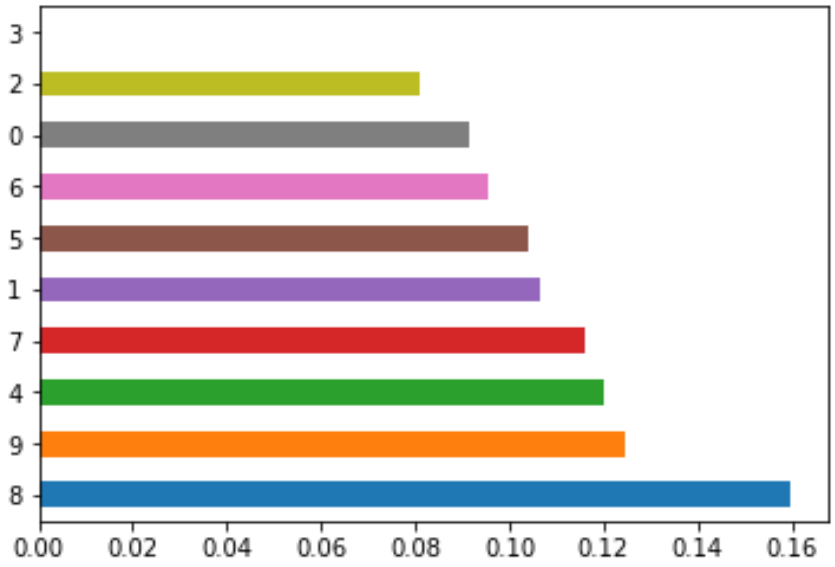

Figure 4: Feature importance graph for the author with ID 'A1M5ZT35YX6TIN'

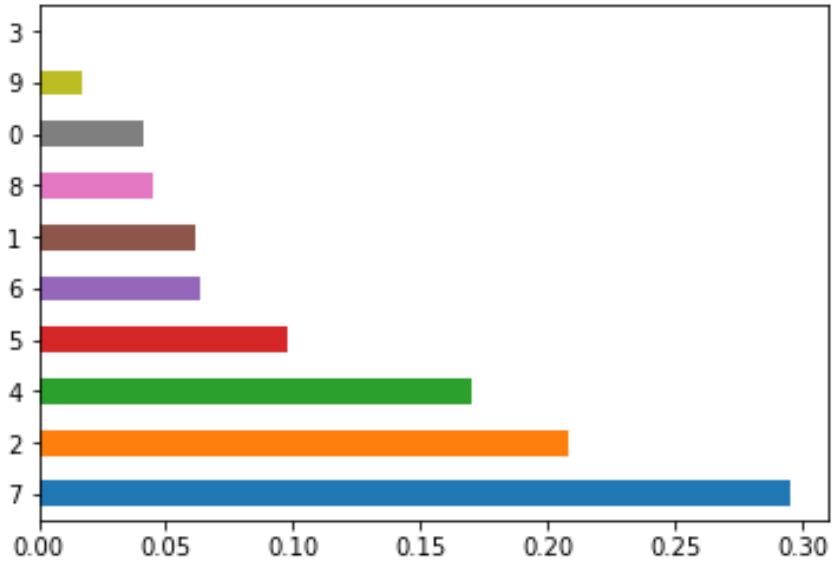

Figure 5: Feature importance graph for the author with ID 'A328S9RN3U5M68'

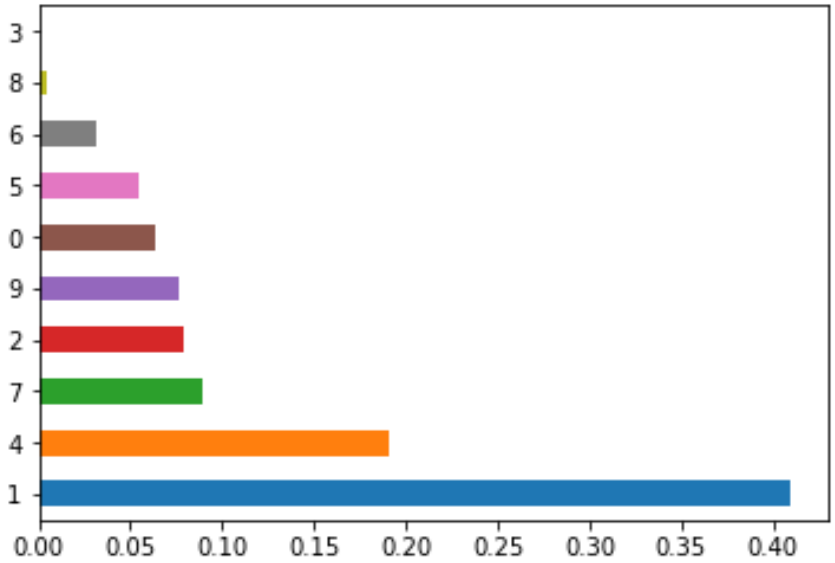

Figure 6: Feature importance graph for the author with ID 'A2NJO6YE954DBH'

As it is evident from the above figures, it is hard to obtain a generalized feature importance method for all the authors.

We implemented a dynamic chi square feature selection method to obtain the top 5000 features for each verifier and train and test the model using those 5000 features. This is not a generalized model as the top 5000

features for different verifiers are different. The result that we got by performing this experiment is shown below.

Accuracy = 0.9741
Precision = 0.9777
Recall = 0.9741
F1-score = 0.9739

7. Conclusion and Future Work

We have proposed a method to perform authorship attribution on reviews in large-scale datasets. This method employs individual author verifiers to solve the problem of authorship attribution on a large set of authors. The reason why our model works well on large number of authors is that our individual author verifiers are working well. When we combine the results of these verifiers, which are performing well, we get a good collective result. Future work that we plan on working includes refining the average word frequency class feature and work on a generalized mechanism to obtain the most important features across all the authors.

References

1. http://www2.cs.uh.edu/~arjun/papers_new/Shrestha%20et%20al.%20CICLING%2016.pdf
2. http://www2.cs.uh.edu/~arjun/courses/ml/Projects.pdf
3. https://pypi.org/project/textstat/
4. Meyer zu Eissen, S., Stein, B.: Genre classification of web pages. In Biundo, S., Frhwirth, T., Palm, G., eds.: KI 2004: Advances in Artificial Intelligence. Volume 3238 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2004) 256–269
5. https://pan.webis.de/clef12/pan12-web/author-identification.html
6. https://pan.webis.de/clef13/pan13-web/author-identification.html
7. http://www.gutenberg.org/ebooks/search/%3Fsort_order%3Ddownloads
8. https://scikit-learn.org/stable/
9. https://towardsdatascience.com/
10. https://www.stackoverflow.com/
11. https://machinelearningmastery.com
12. https://www.mysql.com/downloads/
13. https://dev.mysql.com/downloads/connector/python/