Author Identification Using Text Snippets

Suryank Tiwari[1] **Prateek Agarwal**[2]
Indraprashtra Institute of Information Technology, Delhi
1. suryank19019@iiitd.ac.
2. prateek19070@iiitd.ac.

Abstract

This project deals with author identification given short text snippets by using stylometric and lexicographical features. The goal is to construct a textual dataset that highlights an author's characteristics well. Further, analysis and extraction of said author specific attributes and behaviours from this dataset to arrive at a model which generalizes well over identifying authors from short text snippets. This serves as a good platform for the applications such as plagiarism detection, writers of threatening documents, identification of change of author in the midst of writing etc. to name a few.

1 Problem Definition

Each of us has their own different writing style and vocabulary. The features that can be extracted from a piece of text that characterize an author are called stylometric features. These can be used to identify the author uniquely by their work. However, the length of the text and the number of authors considered at a time also matter. Here we present an analysis on the same.

With each keystroke, each author imparts themselves unto their work; most of this is subconscious. We propose to train a machine learning model on short text snippets to leverage these properties and identify the author. The task of recognizing authors accurately from textual snippet analysis is hard for humans as well.[2]

This work can serve as a good platform for understanding or simulating such human behavior, plagiarism detection, identification of change of author in the midst of texts, consistent guided writing tools and identification of author in resolution of disputes of historical documents, etc.

Apart from author specific features, we also take in account lexicographical features which may involve a person's vocabulary and other indirect tendencies that may help solve the problem.

2 Importance of the Project

The importance of the project becomes apparent from the effect of applications that can use this work as foundation.
Some of the various applications that may follow this concept are:

- Simulating/Mimicking author behavior by machines.
- Identifying plagiarism, author changes, author claims out of their works.
- Better information retrieval systems.
- Tone, delivery and message consistency guidance by automated systems like Grammarly.

3 Data sets

We take inspiration from a Kaggle competition: Spooky Author Identification. The dataset generated in turn also follows the same structure, so similar statistical analysis tools, and machine learning models.

The baseline dataset is available at:

https://www.kaggle.com/c/spooky-author-identification/data

This Kaggle dataset consists of work of three authors in the form of solitary sentences paired with the corresponding author. There are a total of 36537 training instances and 9135 testing instances present. Following the trend and structure of this

dataset, we have created our own dataset by web scrapping.

To obtain dataset of various authors we have scrapped all author data from the American Literature website:

https://americanliterature.com/authors

Following are the properties of the data that was collected:

- The works scraped range over 415 authors, totalling over 9416 document works in stories, poems and plays.
- Total data obtained after scraping the mentioned website is 398 MBs in size.

This data needs to be pre-processed and structured to be of any use. Following are the steps taken to arrive at the data set:

- Pre-processing of author works obtained to remove miscellaneous and junk data.
- Sentence tokenization and mapping author snippet pairs to form a dataset.

After scraping, the question while creating the data set was to evaluate which sentences to include for an author and which don't. We needed to find sentences that are common and sentences that are unique to the authors.
To achieve this the following strategy was considered:

- Creating a list of number of sentences an author has, out of this we cherry picked three authors that were close in their statement counts. We create a small corpus of these authors.
- Pre-processing the corpus set, in terms of tokenization, lemmatization, punctuation removal and case folding.
- Removing unnecessary sentences collected while web scraping. This was done by creating a list of triggers that was generally seen after scraping.
- Sentences that consisted less than 5 words were removed to keep the sample length at a certain length.
- This pre-processed data was converted to features using a count vectorizer which were then passed through a Multinomial Naive Bayes Model. An author identification accuracy of 85.96% was observed.
- We get the class wise probabilities for each sentence. Then using the sentence IDs we create a new data frame which contains unprocessed sentences corresponding to the actual class probability obtained from Naive Bayes.
- The higher probability a sentence has, the more confident the model is about it belonging to a certain class. Therefore these sentences are considered to be unique. Lower probability sentences are considered to be common.
- We take 90% of the top sentences to remove bad outliers and take a 70-30 ratio of common to unique sentences to select sentences of an author.
- A data set which contains corresponding selected sentence-author pairs is generated.
- This process is repeated for each author and resultant data is split into test and train randomly. Dataset creation is completed.

Following is the class sample count distribution of the data set created:

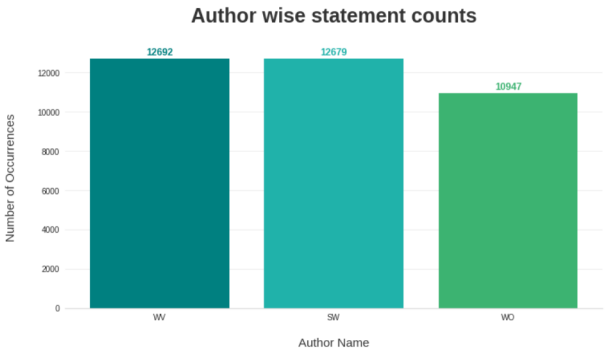

Figure 1: Class Distribution

Where SW, WV and WO are the famous authors Shakespeare William, Woolf Virginia Allan and Wilde Oscar. The number of samples are nearly equal in two classes and there is a slight data imbalance in the third class.

The dataset insights are provided in greater detail in the images that follow.

Document Length Statistics(in characters)

Author/Property	Shakespeare	Woolf	Wilde
Min Doc Length	702	1578	94
Max Doc Length	194029	946911	431823
Average Doc Length	12929.10	163153.61	37320.36

Figure 2: Document Length Statistics

Sentence Length Statistics(in characters)

Author/Property	Shakespeare	Woolf	Wilde
Min Sentence Length	0	0	0
Max Sentence Length	1568	1158	2540
Average Sentence Length	91.57	100.27	79.59

Figure 3: Sentence Length Statistics

Word Length Statistics(in characters)

Author/Property	Shakespeare	Woolf	Wilde
Min Word Length	0	0	0
Max Word Length	456	50	29
Average Word Length	4.19	4.44	4.16

Figure 4: Word Length Statistics

Following is the TSNE plot of the data after using a simple tf-idf vectorizer to generate features:

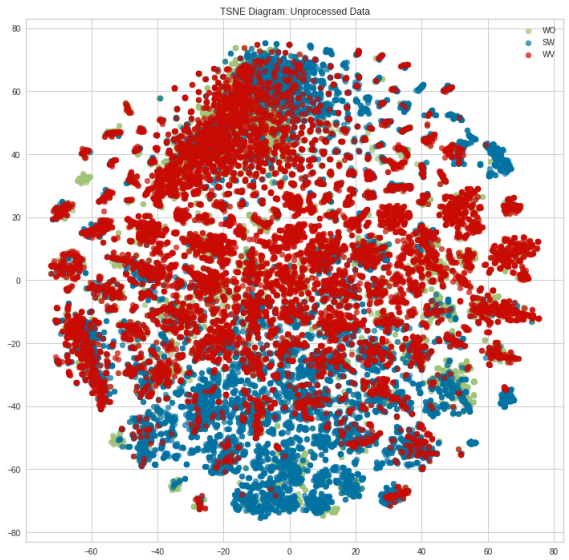

Figure 5: TSNE Plot - Unprocessed Data

and here is the UMAP plot of the same features:

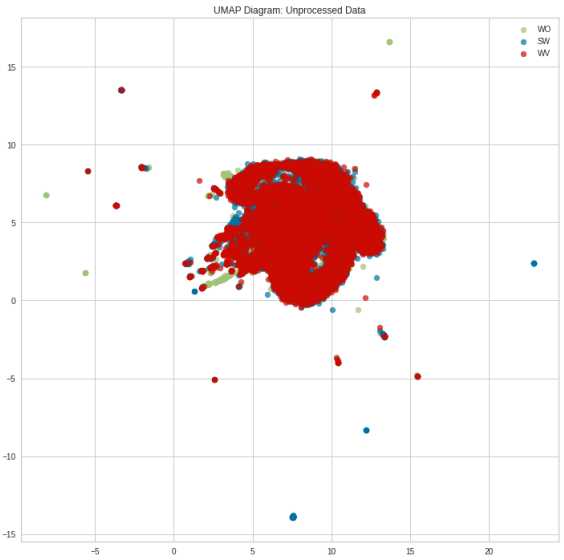

Figure 6: UMAP Plot - Unprocessed Data

Following is the feature diagram for POS Tag counts across the final dataset generated.

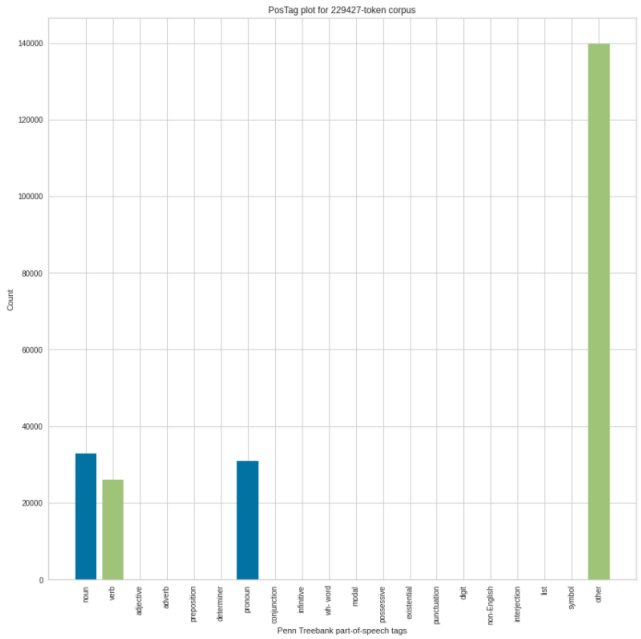

Figure 7: POS Tag Plot

So, the dataset consists of 3 authors, 36319 training samples and 9081 testing samples in total. The learning model we create and the baseline will be evaluated on this data set created.

4 Baselines

The performance reached on Kaggle are state-of-the-art on the provided data set. Not all the entry

notebooks are made public but a good number of them are. Comparison of this project's performance with the public models will be performed on the data set for a structured analysis.
The top publicly available baseline model that is being considered is the following Kaggle notebook:

Baseline Notebook

The above model obtained a multi-class log loss of 0.326 on the Kaggle dataset included in section 3. The baseline model results on our dataset are included in the results section.

5 Proposed Solution Architecture

5.1 Evaluation Metric

Following in the fashion of the Kaggle event, and considering the multitudinal nature of the classes, the evaluation metric for our problem is the multi-class log loss.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

where N is the number of observations in the test set, M is the number of class labels (3 classes), *log* is the natural logarithm, y_{ij} is 1 if observation i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j.

However, we are also presenting the classification report on the techniques present in the latter sections.

5.2 Preprocessing

The text data obtained is in raw format which needs to be preprocessed. These techniques include:

- Tokenization - The sentences present in the author text are tokenized to generate a stream of tokens.
- Lemmatization - Lemmatization is a process of producing the root word out of the word present in the text. After the tokens are produced, each word is then brought to the lemmatized form.
- Stopword Removal- Stopwords need to be removed to generate meaningful features.
- Contraction Expanding - Various contractions present in the text data need to be expanded.
- Punctuation Removal - Punctuations need to be removed to better assess the text data.
- Lowercase conversion- Words present in different cases need to be brought to a standard case.

After the preprocessing, the dataframe of a list of tokens for each sentence is obtained to be processed upon further.

5.3 Features

Upon applying some statistical metrics to the dataset we had generated, we obtain some features such as number of words being used by the author and number of punctuations used by the author. Following are the violin plots of these features:

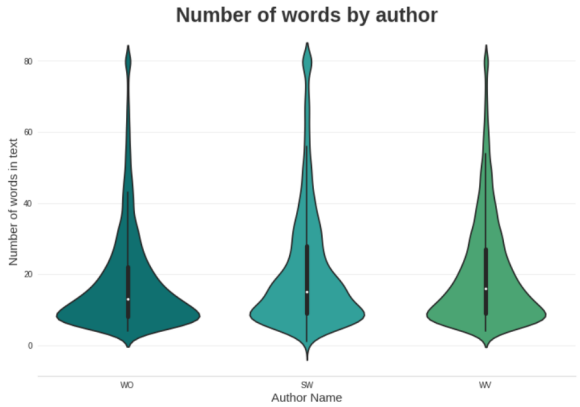

Figure 8: Words By Author

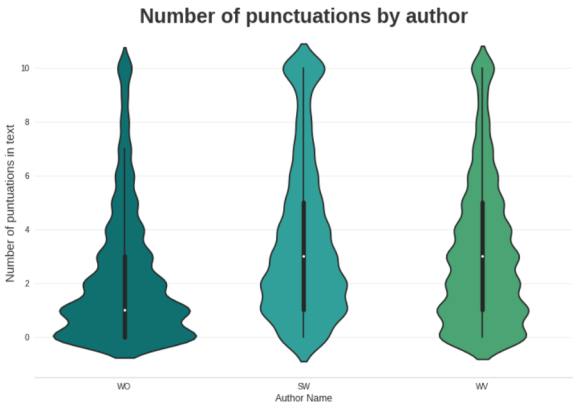

Figure 9: Punctuations By Author

The violin plots depict a portion of the features that were used to follow for author classification. Various author and text dependent features that

were extracted, analysed and tested to fit a suitable machine learning model are:

- Syllable Count
- Character Count
- Word Count
- Average Syllable Length
- Average Word Length
- Unique Word Count
- Punctuation Frequency
- Stop Word Frequency
- Vader Intensity Features (4)
- Word Counts by Length (16)

The features mentioned above are stylometric in nature, aside from the Vader Intensity Features which are sentiment analysis features. Aside from the above features, these techniques were also used to generate features in bulk:

- **Tf-idf Vectorizer:** Generating TF-IDF features from the preprocessed data with 'n-gram' range 1 to 3. Several thousand features are generated from this method which are brought to a lower level by using **Single Valued Decomposition (SVD)**.
- **POS Tag Tf-idf Vectorizer:** Converting preprocessed sentences to POS tag form and then using a TF-IDF Vectorizer to generate several thousand features. These are again converted to a lower number using SVD.
- **Count Vectorizer:** Obtaining a matrix of token counts.

These features are not directly stylometric but indirectly help the cause as they help capture the vocabulary richness and contain the patterns among word, structure and frequency usage an author makes.
The above features are combined into two feature sets: One feature set contains all the features (feature set 1) and the other one contains all the stylometric features and the TF-IDF Vectorizer features only (feature set 2).
Single Valued Decomposition was used to compress the highly sparse information that was obtained from Tf-idf vectorizers into 20 features.

Here is the TSNE visualization of the feature set 1:

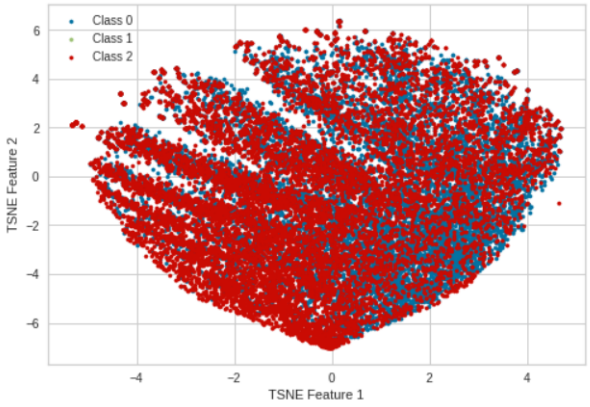

Figure 10: TSNE Plot - Feature set 2 (SVD)

The TSNE visualization of the feature set 2 is:

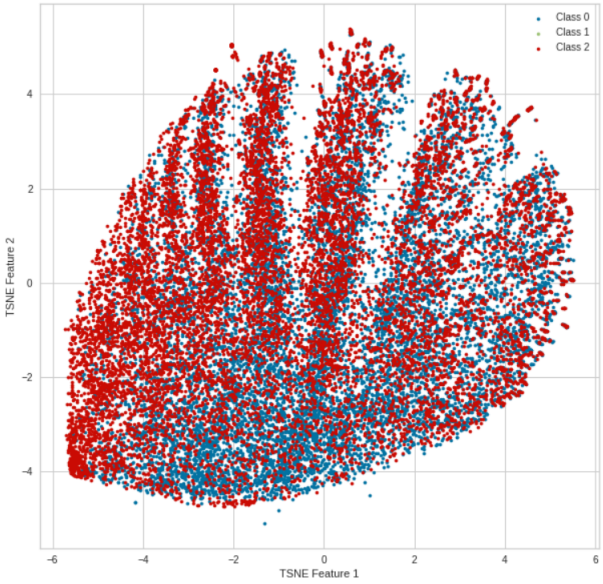

Figure 11: TSNE Plot - Feature set 1

5.4 Machine Learning Models

Following subsections describe the machine learning models used:

5.4.1 Passive Aggressive Classifier

Since passive aggressive classifiers are known to perform on short text data like tweets, we implemented a passive aggressive classifier using sklearn library to train on the second feature set generated.
However, since Passive Aggressive Classifiers do not generate probability predictions, the evaluation metric 'log loss' is not calculate for this machine learning model, only accuracy is computed.

5.4.2 Logistic Regression

A logistic regression pipeline which handles the imbalance in the number of samples per class and selects features is followed by a class weight balanced Logistic Regression module. The second set of features generated is used for this machine learning model as well.
Multiclass Log loss and accuracy report is generated and is presented in the results section.

5.4.3 MultiNomial Naive Bayes

This machine learning model is trained on the first feature set and is the best performing model. MultiNomial Naive Bayes from sklearn is applied for this problem.
Multiclass Log loss and accuracy report is generated and is presented in the results section.

6 Literature Review

Rexha, Kröll, Ziak and Kern put human beings to the test on the problem of author identification on documents with high content similarity.[2] They noted an inter rated agreement of 28 percent with a confidence of 72 percent that the results were genuine and not random. Their results clearly show that this problem of author identification with high content similarity is not easy for humans to perform as well.

Brocardo, M.L. et al. worked on the email content and extracted user profiles from n-grams from sample documents and then calculating a threshold value for the verification phase.[1] The study used only one type of features out of many available stylometric features. The results are corresponding to the metric Equal Error Rate which has the value of 14.35 percent for 87 users.

Abbasi and Chen[3] worked on extracting linguistic features from online message boards to identify and locate patterns of terrorist communication The models applied were SVM and Decision tree in which SVM outperformed the decision tree.

7 Results

The web scraped data of the authors for their various works were transformed into structured sentences. These sentences were then fed into the machine learning models, accuracy and multiclass log loss values were obtained.

These results were obtained on 70:30 ratio of common and unique sentences for the specified authors in the dataset section.

Performance Report of Logisitic Regression Model:

	precision	recall	f1-score	support
SW	0.84	0.87	0.85	3144
WO	0.71	0.72	0.71	2786
WV	0.80	0.76	0.77	3150
accuracy			0.78	9080
macro avg	0.78	0.78	0.78	9080
weighted avg	0.78	0.78	0.78	9080

Figure 12: Logistic Regression Classification Report

From the classification report of Logistic Regression, we can infer that the author SW is classified with much accuracy and author WO is classified with least accuracy, hence the overall accuracy is 78%.

Performance Report of Multinomial Naive Bayes Model:

	precision	recall	f1-score	support
SW	0.89	0.92	0.90	3144
WO	0.82	0.69	0.75	2786
WV	0.78	0.87	0.82	3150
accuracy			0.83	9080
macro avg	0.83	0.83	0.83	9080
weighted avg	0.83	0.83	0.83	9080

Figure 13: Multinomial NB Classification Report

The classification report of Multinomial Naive Byes classifier also shows promise in classifying the author SW with much accuracy and author WO with least accuracy. However, the accuracy is improved as compared to previous model with the overall accuracy of 83%.

Performance of Passive Aggressive Model:

	precision	recall	f1-score	support
SW	0.88	0.88	0.88	3144
WO	0.85	0.52	0.65	2786
WV	0.68	0.91	0.78	3150
accuracy			0.78	9080
macro avg	0.80	0.77	0.77	9080
weighted avg	0.80	0.78	0.77	9080

Figure 14: PassiveAggressive Classification Report

The PassiveAggressive classifier performs poorly for the author WO in comparison with the previous models and performs better than Logistic Regression on the author SW. But, this neutralizes the

accuracy to be as similar as that of the Logistic Regression.

Performance of Baseline Models on our dataset:

Table 1: Baseline Validation Data Statistics

S.No.	Model Name	Log Loss
1.	XGBoost	0.89
2.	Naive Bayes	2.99
3.	Multinomial NB	0.41

Model Performances in terms of Multiclass Log Loss:

Table 2: Test Data Statistics

S.No.	Model Name	Log Loss
1.	LogisticRegression	0.50
2.	Multinomial NB	0.42
3.	PassiveAggressive	

From the results obtained, we can see that the baseline log loss and the log loss obtained using Multinomial Naive Bayes model are comparable. However, the baseline value is on validation data and our model has performed on the test data with gold labels.

We were able to achieve 83% accuracy and multiclass log loss of 0.42 using the Multinomial Naive Bayes model.

Link to the Google Drive for our work:

Google Drive Link

8 Conclusion and Future Works

The results surpass human performance at the task on hand with a total accuracy of 83% overall. The best performing model was Multinomial Naive Bayes model. The performance of the baseline model and our methdology is comparable, however the baseline model finds logloss on validation data and we find it on a much larger test data.

For future work, advanced stylometric coefficients can be computed like John Burrows' Delta Method. Further research my also lead to new devised stylometric features.

9 References

[1] Brocardo, Marcelo & Traore, Issa & Saad, Sherif & Woungang, Isaac. (2013). Authorship verification for short messages using stylometry. Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS). 1-6. 10.1109/CITS.2013.6705711.

[2] Rexha, A., Kröll, M., Ziak, H. et al. Authorship identification of documents with high content similarity. Scientometrics 115, 223–237 (2018). https://doi.org/10.1007/s11192-018-2661-6

[3] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group Web forum messages," in IEEE Intelligent Systems, vol. 20, no. 5, pp. 67-75, Sept.-Oct. 2005, doi: 10.1109/MIS.2005.81.