AUTHOR IDENTIFICATION USING TRADITIONAL MACHINE LEARNING MODELS

Ojaswi Binnani

Language Technologies Research Centre,
International Institute of Information Technology-Hyderabad, India

ABSTRACT

The Internet has many useful resources with bountiful information at our fingertips. However, there are nefarious uses to this resource, and can be misused in cybercrime, fake emails, stealing content, plagiarism etc. In many cases, the text is anonymously written, and it is important to accurately find the author to bring the criminal to justice. The topic of author identification helps with this task, where from a set of suspect authors, the writer of a given text will be determined. We aim to create a computationally non-complex model that works to find the author of a given text. The model will not require as much data as deep learning methods. This paper focuses on the use of various stylometric and word-based features as well as different machine learning models to create a classifier that gives the best accuracy. We find that the XGBoosting algorithm performs this task with a good accuracy.

KEYWORDS

Author Identification, Forensic Linguistics, Machine Learning.

1. INTRODUCTION

The Internet has a vast and large knowledge base that has many advantages in connecting reliable content and real people. However, the Internet has also paved the way for cybercrime such as plagiarism from other legitimate content-creators. In this situation, when it is difficult to detect where the content or text originated from, this task can prove to be useful.

Another scenario is within Forensic Linguistics. A concept called Linguistic Fingerprinting or Write Print theorizes that each person has a unique style of using language. According to Olsson [24], "A linguistic fingerprint is a concept put forward by some scholars that each human being uses language differently, and that this difference between people involves a collection of markers which stamps a speaker/writer as unique; similar to a fingerprint. Under this view, it is assumed that every individual uses languages differently and this difference can be observed as a fingerprint." This concept is very attractive to the law enforcement, but there is hardly any evidence to prove this. It has however been used in cases such as the case against the Unabomber. If the author identification model with high accuracy can say that a piece of work is writ-ten by a specific suspect author, then this can be used as evidence (with other additional evidence) to bring justice.

Author Identification has also been used to credit people with works which were anonymously published. One such example is the disputed Federalist Papers. The Federalist papers were released to the public supporting the constitution of America written by Alexander Hamilton, James Madison and John Jay. It was a series of 85 articles published anonymously. Most of the

David C. Wyld et al. (Eds): NLPA, AIS, BDAP, SOENG, IPPR - 2022
pp. 13-20, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121402

writings were credited without dispute, but 12 of the articles were either written by Hamilton or Madison. Through Author Identification task done by Mosteller and Wallace [23], they concluded that it was Madison who wrote all twelve of the disputed Federalist articles. Other studies [3; 10] also supported this conclusion.

We divide the author identification task into two sub-tasks viz. extracting features that can accurately represent the text and testing various traditional machine learning models to see which works best for this task.

To choose features that represent a given text, we explore the different types of features that exist and examples of each type and choose a subset that works for classifying the author.

For our second task of testing various traditional machine learning models, we set the task as a single-label multi-class classification problem where each author is a single class, and each text can be written by only one author. Then we test different types of algorithms explained further in subsection 6.2.

In section 2 we look at related work that has been done in this area; in section 3 we look at the dataset used. In section 4 we discuss the different types of features that exist. We go into depth of these features in subsection 5.1 and subsection 6.1. Machine Learning models are discussed in subsection 5.2 and subsection 6.2.

2. RELATED WORK

Khan et al. [16] discussed the differences between the three tasks in author analysis - one of which is author identification. The paper discusses different types of features and types of models used in author identification and proposes a method to identify an author of an email.

Zheng et al. [32] looked at authorship identification in online messaging. One of the languages they looked at was English and came up with 270 features categorized into five types: lexical features, word-based features (not similar to our word-based feature classification in section 5), syntactic features, structural features and content specific features. Using these 270 features, Li et al. [19] found that a Support Vector Machine (SVM) outperformed the other models. This proves that a traditional model with sufficient meaningful features can be used in author identification. Mohsen et al. [22]; De Vel [6] also used SVMs for author identification.

Zhou and Wang [33]; Qian et al. [27] worked with three stylometric features: average sentence length, average word length and Hapax Legomenon Ratio and applied Glove Vector Embeddings to news articles. They worked with RNN and LSTM respectively. Their models suffered from an overfitting problem due to the computational complexity of their models and their insufficient dataset.

Iqbal et al. [13] has a list of features, and for any two authors has a subset of features that are applicable to one author and not the other. They call this the write print between two authors, and use these features to classify the author of a malicious email.

3. DATASET

In the paper, we use the Reuter 50 50 dataset [27]. This dataset was developed by ZhiLiu in 2011 and has been used in author identification experiments since then.

The dataset consists of 50 authors' works. Each author has 50 articles in the training set and 50 articles in the testing set. There is no overlap of articles between the training set and testing set. Each of the articles is in the news genre.

For each author, the number of words per article ranges from 468 to 569 words, averaging around 512 words per article. The sizes of the articles range from 2.7 kB to 3.6kB, averaging at 3.13 kB per article.

In this paper, the training and testing sets are combined and then a 90-10 split is done for training and testing.

4. FEATURE TYPES

There are three different types of features that we will use to represent each article in the training set.

4.1. Word-Based Features

These features are based off the words or phrases an author commonly uses. This is very effective when the author commonly uses specific adjectives or writes with a distinctive vocabulary. However, the problem with these types of features is that content words such as words that are only used in the case of a specific topic. An author may be writing on a topic that uses certain jargon, but it is not indicative of the author's personal style. The classification may fail when another author is writing that topic and uses the same jargon.

Some examples of this type of feature include Bag of Words (BOW)[31; 20], Doc2Vec [18], Glove Vector Embeddings [26] and n-grams [14].

4.2. Stylometric-Based Features

Since the previous feature may not be accurate on its own, we use stylometric features as well. These features aim to be similar for a single author despite the different topics of the author's texts. Vocabulary richness scale can be one such feature since the variety of vocabulary has little correlation with the topic of the text. Another feature can be the average sentence length or average word length. Both features refer to the author's choice in sentence structure or vocabulary but not necessarily to the topic of the text.

Stylometric features can be

- article-level e.g. number of paragraphs, number of sentences, number of words
- paragraph-level e.g. number of sentences in a paragraph, average length of sentence, number of words in a paragraph
- word-level e.g. number of small words (less than four characters), average length of words, frequency of each alphabet
- vocabulary richness e.g. Hapax Legomenon Ratio

4.3. Syntax-Based Features (Punctuation)

Syntax features can refer to the usage of function words such as determiners and auxiliary verbs. In this paper the features we focus on are punctuation features. Punctuation features can be used

for author identification because certain punctuation can be a differentiator between authors. For example, some authors may use semicolons in between sentences and others may not. The number of commas in a sentence can be another distinguishing feature.

Examples of syntax-based features are the count of specific function words e.g. ‘between’, ‘a’, ‘nor’. Zheng et al. [32] found 150 function words that can be used as features. Another type of syntax-based features is the count of different punctuation e.g. exclamation points, commas, semi colons. Zheng et al. [32] found 8 features.

5. EXPERIMENTS

Zhou and Wang [33]; Qian et al. [27] implemented three stylometric features (average word length, average sentence length, and Hapax Legomenon Ratio) through traditional machine learning models such as Support Vector Machine, Gradient Boosting etc. gets a maximum accuracy of 12%.

To further increase the accuracy of authorship identification, rather than using computationally complex machine learning models such as RNNs and LSTMs (the methods implemented by Zhou and Wang [33]; Qian et al. [27]), more features were included to enhance the traditional models.

5.1. Feature Extraction

In Section 5, we discussed the various types of features that exist. To train our models, we need to select a viable subset from each type of feature for the task of author identification.

5.1.1. Word-Based Features

Doc2Vec [18] is the primary word-based feature in this paper. Doc2Vec is a modification to Word2Vec [21] and assigns a vector to each article in the dataset. The vectors represent the words in the article, the ordering of the words as well as their location in paragraphs. These vectors are numerical values that can be fed into a machine learning model.

5.1.2. Stylometric-Based Features

We use the features used by Zhou and Wang [33]; Qian et al. [27] (average word length, average sentence length and Hapax Legomenon Ratio) and the length of the article in words and sentences. The Hapax Legomenon Ratio is the ratio of the number of words used only once in the text to the total number of words in the text.

5.1.3. Syntax-Based Features

We use punctuation features in this paper. The usage of quotation marks, commas, dashes and exclamation marks in an article are counted and used as features in the model.

5.2. Machine Learning Models

Various machine learning models were tested using the above-mentioned features. They are vaguely classified into four types: Linear, Trees, Neighbours and Boosting.

5.2.1. Linear Machine Learning Models

Naive Bayes [29; 15] is a classification technique that often doesn't result in highly accurate outputs However, this is one of the simplest models to run, perfect for a baseline. Both Linear Regression [17; 2] and Support Vector Machine (SVMs) [25; 7; 30] using a linear kernel attempt to linearly separate the classes. Other kernels are possible with the SVM such as polynomials and sigmoids, however, in this task, the linear kernel works the best.

5.2.2. Neighbour-Based Classification Models

K-Nearest Neighbours (KNN) [5] classification is used in this paper. This classification technique is based on the Nearest Neighbour Algorithm. The main downfall of this method is that the results may be inaccurate if the dataset is skewed, i.e. one class has more data points than the other. The dataset we are using is balanced, and hence this method has the potential to decently classify the works.

5.2.3. Tree-Based Classification Models

Tree Based Machine Learning models are found to work decently with discrete models (non-quantifiable results) such as this task. They are also found to be good at capturing complex, non-linear relationships between the classes, hence if the linear models do not produce a desirable output, it is likely that the tree models will. Random Forest [4], Extra Trees [11] and Decision Trees [28] are used.

5.2.4. Boosting Models

Boosting has proven to be a useful machine learning method due to its speed and less complexity. It is a model based on using several weak models working in tandem to become a strong learner. Boosting, like tree-based models, are good at capturing complex, non-linear relationships and are good at discrete classification. AdaBoost (Adaptive Boosting) [8] and XGBoost (a variation on Gradient Boosting) [9; 1] are used in this paper.

6. RESULTS

6.1. Features

Table 1. Feature Importance – Ranking of Features from most to least important excluding Doc2Vec Positions

Average Sentence Length
Average Word Length
Number of Commas
Number of Dashes
Number of Quotation Marks
Hapax Legomenon Ratio
Number of Words
Number of Sentences
Number of Exclamation Marks

Table 1 represents all the features and their importance in the XGBoost Model. The least important features are the number of exclamation points and certain positions in the Doc2Vec

vectors, while the more important features are the stylometric and syntax-based features as well as certain Doc2Vec vector positions.

We can see that the stylometric features contributed heavily to a good classification. The average sentence length proved to the most important feature within the whole group of features.
We can also see that the punctuation features (syntax features) are important with the usage of commas being the most important feature within this set.

6.2. Machine Learning Models

Table 2. Results of Various Machine Learning Models

Model	Training Accuracy	Testing Accuracy	F1 Score	Recall	Precision
Naïve Bayes	0.23	0.16	0.114	0.092	0.164
Liner Regression	0.42	0.36	0.330	0.333	0.344
SVM	0.59	0.48	0.463	0.464	0.486
K-Nearest Neighbour	1.00	0.26	0.246	0.247	0.267
Random Forest	1.00	0.71	0.716	0.731	0.735
Extra Trees	1.00	0.68	0.694	0.705	0.697
Decision Trees	1.00	0.57	0.577	0.578	0.588
AdaBoost	0.90	0.76	0.770	0.780	0.773
XGBoost	1.00	0.83	0.826	0.830	0.842

Table 2 shows the various scores for the model using the same feature set. As predicted, Naive Bayes performs the worst with a F1 score of 0.114. We can see all the Linear Models and the K-nearest Neighbour model performed with less than 50% accuracy which makes them worse than a weak model.

The Tree models performed with higher than 50% accuracy with Random Forest being the best within the Tree algorithm group with a F1 score of 0.716.

The Boosting models performed better than all other groups of models, XGBoost performing the best with an F1 score of 0.826. AdaBoost performed second best amongst all the algorithms with a F1 score of 0.770.

From these results we can see that the relationship between each of the classes is non-linear. All Linear classification models failed to classify the testing articles, and even failed to classify the training articles since they had an abysmal training accuracy. The models that are good at classifying non-linear data points such as Tree-Based Models and Boosting Models clearly perform well.

7. CONCLUSIONS

In this paper, we aimed to use traditional machine learning models to do the task of author identification using three types of features: word-based, stylometric and syntactic. We prove that

all three of these features are useful in author identification by seeing their importance in the XGBoost model. We also see that Boosting Models perform the best amongst the traditional machine learning models since both Boosting models outperform the remaining models. XGBoost performs the best with 0.826 F1 score and 82.5% testing accuracy. This result is found due to the non-linear relationship between the classes. Hence, this is consistent with our hypothesis that traditional machine learning models can be used in author identification, which reduces the complexity to perform. It can work with an equal or better accuracy than that of complex machine learning models.

REFERENCES

[1] Bradly Boehmke and Brandon Greenwall. 2020. Chapter 12 Gradient Boosting. CRC Press.
[2] Bradly Boehmke and Brandon Greenwall. 2020. Chapter 4 Linear Regression. CRC Press.
[3] Robert A. Bosch and Jason A. Smith. 1998. Separating hyperplanes and the authorship of the disputed federalist papers. The American Mathematical Monthly, 105(7):601–608.
[4] L Breiman. 2001. Random forests. Machine Learning, 45:5–32.
[5] Padraig Cunningham and Sarah Delany. 2007. k-nearest neighbour classifiers. Mult Classif Syst.
[6] Olivier De Vel. 2000. Mining e-mail authorship.
[7] Theodoros Evgeniou and Massimiliano Pontil. 2001. Support vector machines: Theory and applications. volume 2049, pages 249–257.
[8] Yova Freund and Robert E. Schapire. 1995. A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting. In Second European Conference on Computational Learning Theory (EuroCOLT-95), pages 23–37.
[9] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5):1189 – 1232.
[10] Glenn Fung. 2003. The disputed federalist papers: Svm feature selection via concave minimization. pages 42–46.
[11] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. Machine Learning, 63:3–42.
[12] John Houvardas and Efstathios Stamatatos. 2006. N-gram feature selection for authorship identification. volume 4183, pages 77–86.
[13] Farkhund Iqbal, Rachid Hadjidj, Benjamin C.M. Fung, and Mourad Debbabi. 2008. A novel approach of mining write-prints for authorship attribution in e-mail forensics. Digital Investigation, 5:S42–S51. The Proceedings of the Eighth Annual DFRWS Conference.
[14] D. Jurafsky and J.H. Martin. 2014. Speech and Language Processing. Always learning. Pearson.
[15] Pouria Kaviani and Sunita Dhotre. 2017. Short survey on naive bayes algorithm. International Journal of Advance Research in Computer Science and Management, 04.
[16] Sobiya Khan, Smita Nirkhi, and Rajiv Dharaskar. 2012. Author identification for e-mail forensic. Proceedings of National Conference On Recent Trends In Computing NCRTC.
[17] Khushbu Kumari and Suniti Yadav. 2018. Linear regression analysis study. Journal of the Practice of Cardiovascular Sciences, 4:33.
[18] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. 31st International Conference on Machine Learning, ICML 2014, 4.
[19] Jiexun Li, Rong Zheng, and Hsiu-chin Chen. 2006. From fingerprint to writeprint. Commun. ACM, 49:76–82.
[20] Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. 2018. Bag-of-words as target for neural machine translation. pages 332–338.
[21] Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. pages 1–12.
[22] Ahmed M. Mohsen, Nagwa M. El-Makky, and Nagia Ghanem. 2016. Author identification using deep learning. In 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 898–903, Los Alamitos, CA, USA. IEEE Computer Society.
[23] Frederick Mosteller and David L. Wallace. 2012. Applied Bayesian and Classical Inference: The Case of The Federalist Papers. Springer Series in Statistics. Springer New York.
[24] John Olsson. 2008. Forensic Linguistics: An Introduction to Language, Crime and the Law. Bloomsbury Publishing.

[25] Edgar Osuna, Robert Freund, and Federico Girosi. 1970. Support vector machines: Training and applications. Tech Rep A.I. Memo No. 1602.
[26] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543.
[27] Chen Qian, Ting He, and Rao Zhang. 2017. Deep learning based authorship identification.
[28] J. R. Quinlan. 1986. Induction of decision trees. Mach. Learn., 1(1):81–106.
[29] Irina Rish. 2001. An empirical study of the naïve bayes classifier. IJCAI 2001 Work Empir Methods Artif Intell, 3.
[30] Konstantinos Veropoulos, N. Cristianini, and C. Campbell. 1999. The application of support vector machines to medical decision support: A case study. Adv Course Artif Intell.
[31] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics, 1:43–52.
[32] Rong Zheng, Jiexun Li, Hsiu-chin Chen, and Zan Huang. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. JASIST, 57:378–393.
[33] Liuyu Zhou and Huafei Wang. 2016. News authorship identification with deep learning.