Enron Email Classification

P28: Jackson Ingraham, Sam Smith, Corey Capooci
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
{jtingrah, scsmit23, cvcapooc}@ncsu.edu

1 Background

1.1 Dataset

The Enron Corporation was an American energy company which committed one of the largest instances of accounting fraud in history. The dataset we are using is a corpus of around 500,000 emails from around 150 users, mostly senior management of Enron [3]. This data was originally released to the public by the Federal Energy Regulatory Commission during its investigation of Enron. The dataset has since been modified slightly, correcting some integrity issues and redacting certain emails.

1.2 Problem Statement

Authorship attribution is the task of identifying the author given a document that they have written [6]. Before computational means of determining authorship, experts studied the writing style of individual writers in order to determine the true author of an anonymously written text. Today, machine learning and natural language processing have become effective tools for solving this problem.

For this project, we look at the Enron emails in order to decide which employee wrote the email. We attempt to improve upon the latest techniques and ideas in authorship attribution through the experimentation of unique feature sets over an array of different classifiers. The intent is to find new and interesting ways to improve email authorship attribution through machine learning. This project also analyzes the challenge of short-text classification, as the emails we work with are relatively short documents.

1.3 Literature Survey

1.3.1 Authorship Attribution Applications

Authorship attribution (AA) has many real world applications. AA serves a role in academic integrity by offering plagiarism detection [10]. AA is a tool to help solve and deter criminal activity. AA offers copyright detection, and author attribution for malicious code and malicious communication [1]. AA provides detection capabilities for spam, phishing emails, user impersonation, and fake news [4, 11]. As discussed, AA provides enforcement officials whether they are professors or law enforcement officers the ability to uncover participants in fraudulent activity.

It must be noted that uncovering anonymous writers may have negative impacts as well. Writers who intend to hold power accountable may be in danger if their identities are exposed. Anyone developing authorship attribution should understand the possible consequences of providing these techniques to people who intend to use it against innocent individuals.

1.3.2 Features

Feature selection is one of the most important decisions in the success of an AA model. Features are divided into many groups such as structural, syntactic, and linguistic.

Structural or lexical features focus on certain structures of writing and may be broken down further into document-based, word-based, and even character-based features [10]. The document-based features focus on characteristics that describe the document as a whole such as sentence complexity, and the division of text into paragraphs [10]. Word-based features include average word length, vocabulary richness, and word frequency distribution. Character-based features include character frequency distribution [10]. Akin to character-based and word-based features are the word and character chain feature [10]. Chain features track the multiple word or character sequences throughout a document. The lexical features are convenient because they are easily extracted.

Syntactic features involves evaluating the sentence structural of the author and use it during the attribution. These features requires Parts of Speech Tagging (POST) in order to identity the parts of speech within the document. Syntactic features take the quantitative data after POST to find patterns for individual authors. These features may be effective when combined with lexical features, but on their own are shown to be ineffective [1].

Semantic features focus more on the text content, meaning, and vocabulary. These features are more difficult to understand and quantify even with the help of an NLP tool [1].

Linguistic features are defined by the unique stylistic features or errors made by an individual. Some of these features include spelling mistakes, grammatical inconsistencies, and stylistics [2]. Experts in authorship verification use linguistic features to perform their analysis [2]. These features appear to have some practicality and use in machine learning models especially if there is history of their use. The downside is that it may be difficult to find enough unique attributes of writer in order to discover a unique identifying attribute in every text, especially if these texts are short.

Machine learning techniques, such as support vector machines and random forests, are used to create stylometric models for author attribution [7]. Lexical features are more widely used, especially in shorter text documents where it is more difficult to understand the writer's style. As a substitute for classifying text documents based on style, [1] suggests the use of many features to fully capture an author's writing style followed by using weights to determine how much each feature factors into classification.

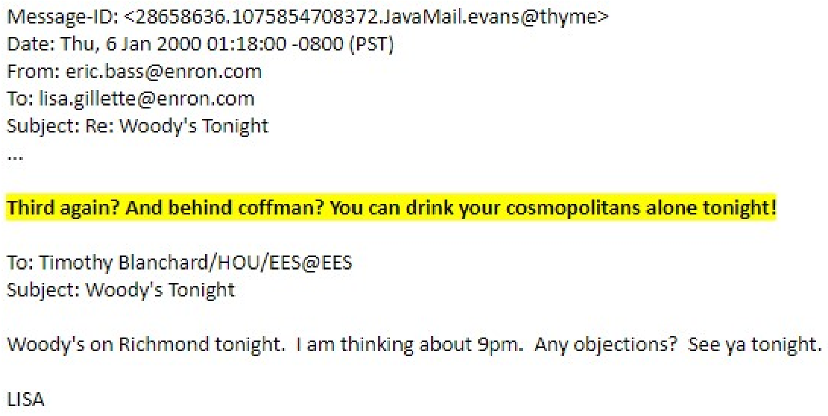
There are many paths to take when determining features for AA and it may be difficult to parse through the literature to determine all of them and their general effectiveness. The ability for a model to determine the most effective features for a problem is crucial to its success. A critique on the AA literature is that it is difficult to determine how each paper determines its features since it appears there are no standard classes of feature sets for written text. This is the main focus of our project.

1.3.3 Classifiers

Another important factor for solving AA problems is the choice of the machine learning model. On longer publications in the PLOS.org repository, [5] tried an unsupervised and a supervised model. For the unsupervised method, a k-means clustering algorithm was deployed using stylometric markers like lexical features. For supervised, a long short term memory (LSTM) model was used and provided better results. [7] employed support vector machines (SVMs) and random forests to conduct a stylometric analysis in order to conduct AA. [8] tested five different classification methods in order to determine the best performers. The models included Delta, K-nearest neighbor, support vector machine, nearest shrunken centroids, and regularized discriminant analysis with Delta being designed specifically for AA problems. [8] determined that nearest shrunken centroids and regularized discriminant analysis were the best performers.

This research provided a great baseline for determining which models perform best, but it may not translate to all types of authorship attribute considering the PLOS.org repository consists of article length texts. These features were also stylometric which are commonly used in authorship attribution problems today, but this research may not be as useful if these features become obsolete.

Overall, there are many factors that go into creating an effective authorship attribution machine learning model, including classifiers and feature selection. Previous work provides commonly used



Message-ID: <28658636.1075854708372.JavaMail.evans@thyme>
Date: Thu, 6 Jan 2000 01:18:00 -0800 (PST)
From: eric.bass@enron.com
To: lisa.gillette@enron.com
Subject: Re: Woody's Tonight
...

Third again? And behind coffman? You can drink your cosmopolitans alone tonight!

To: Timothy Blanchard/HOU/EES@EES
Subject: Woody's Tonight

Woody's on Richmond tonight. I am thinking about 9pm. Any objections? See ya tonight.

LISA

Figure 1: Example of a cleaned email. Only the highlighted portion which the user has written themselves is considered for text classification.

classes of features to provide a good baseline such as lexical features. It also illustrates possible new and interesting features to evaluate, such as linguistic features. In addition, there has been lots of work around identifying the best classifiers for authorship attribution. While this information is help, it may not directly translate this papers work. Lastly, authorship attribution may apply to many problems, but those conducting research in this field must understand the impacts of building models that may impact society for the worse.

2 Method

In this project, we are classifying emails of Enron employees by who wrote them. This process involves filtering and cleaning textual data, engineering useful features, and implementing classification models.

2.1 Data Preparation

As with most data modeling projects, we need to preprocess our data before we can perform any analysis. This involves cleaning and preparing the data for model integration. First, we need to clean the email data since we are only interested in the main email body that the user has written themselves. All text before the body (e.g. date, recipients, subject) and all text after (typically forwarded emails) is filtered out, as these were not necessarily written by the user who sent the email. An example of a cleaned email is shown in Figure 1.

Another worry is the amount of data we are working with. Figure 2 shows how the 500,000 emails are distributed over around 150 users. We notice a large imbalance in the data - many users have few to no sent emails in the dataset, which complicates the multi-classification problem. In order to simplify our initial problem, we only considered the users which have sent above 2000 emails. This leaves 18 users, which is significantly more manageable than the original 150. Figure 3 shows the new class distribution of our reduced dataset.

Another consideration of ours is the length of each email. We observe the distribution of email lengths for our top two users, shown in Figures 4 and 5. Both distributions are right-skewed, with many of these emails being extremely concise. These shorter emails are very difficult to classify, as there is little text to work with. Therefore, we decide to only consider emails which contained a number of words above a chosen threshold of 5 words.

2.2 Feature Engineering

We need to extract numerical features from our textual data to input into the classification models. There are many types of features to extract from text, and the best choice is often problem-specific. We focus our experiments on three types of feature sets: TF-IDF, stylometric, and linguistic.

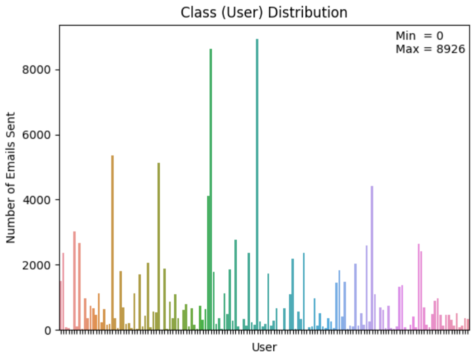


Figure 2: Original distribution of emails

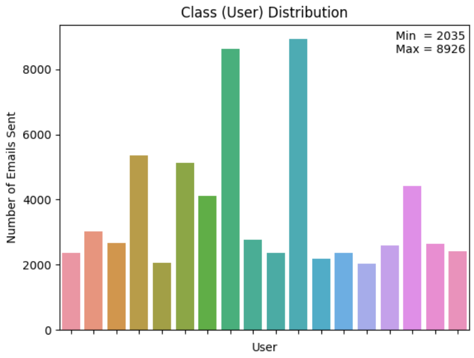


Figure 3: Reduced distribution of emails

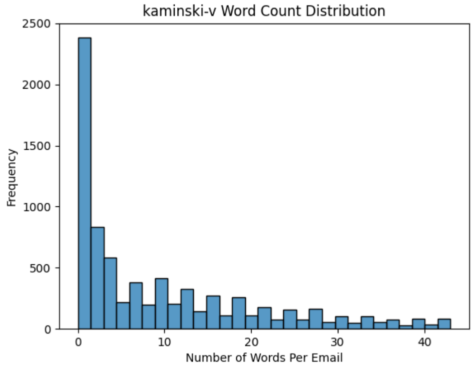


Figure 4: Vince Kaminski's email lengths

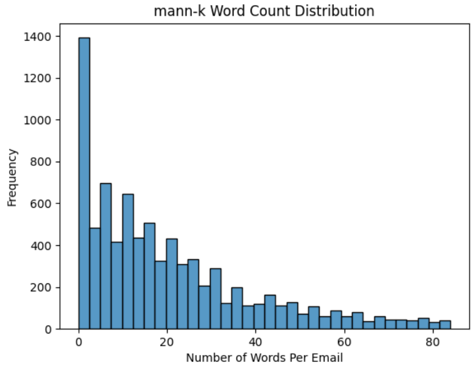


Figure 5: Kay Mann's email lengths

TF-IDF scores are an extremely common and well-researched choice of textual feature. These can be thought of as a way to measure the originality of each word by comparing the number of times the words appears in an email with the number of emails the word appears in. To be exact, the TF-IDF score of a word w in an email e across all N emails E is given by

$$tfidf(w, e, E) = tf(w, e) \cdot idf(w, E)$$

where

$$tf(w, e) = \log(1 + freq(w, e))$$

$$idf(w, E) = \log\left(\frac{N}{count(e \in E : w \in e)}\right)$$

Stylometric features are feature which capture the structure of a person's writing. In our experiments, we extract the following types of features:

- **Lexical-based:** number of alphabetic characters, number of digital characters, number of capital letters, amount of white space
- **Word-based:** average word length, average number of words per sentence
- **Syntactic:** punctuation frequency

Linguistic features are unique stylistic features of a person's writing. Experts in authorship verification use linguistic features as defined by [2] to conduct their work. Therefore, despite being rarely used, linguistic features are possibly a valuable feature for machine learning authorship attribution as well. The linguistic features chosen for this project are geared towards the Enron dataset, specifically towards common features of email discourse. Many linguistic features used in this project found by perusing the dataset. In our experiments, we extract the following linguistic features:

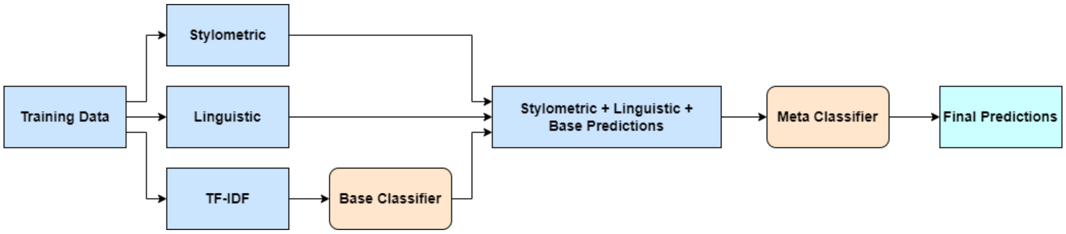


Figure 6: Our ensemble model stacking the TF-IDF labels on the stylometric and linguistic features.

- Whether the email contains elipses ('...')
- Whether the email contains quotation marks
- Whether the email uses common text-based emojis
- Whether the email contains repeated punctuation marks ('!!', '??')
- Whether the email contains numbered lists
- Whether the email contains common greetings ('hi', 'hello', 'greetings', . . .)

2.3 Classification

Once we extract some numerical features from the email texts, we input them into our classification models to establish a baseline on what to expect. We chose to use Random Forest and Linear Support Vector Classification in our initial experiments, which were established to be excellent models for determining authorship attribution by [5], [6], and [7]. Both of these models take a separate approach to categorize: Random Forest creates decision tress attempting to determine the important variables which have the most influence on authorship attribution while Linear SVC attempts to create a linear hyperplane in the dataset. Their diverse methodology makes it worth testing both to find which is better for our dataset.

For control models to establish some sort of baseline, both Logistic Regression and Gaussian Naive Bayes were selected to determine how models unsuited for authorship attribution would perform (as claimed by [5]). For each model, we use a 70/30 training/testing split. Cross-validation is not necessary as we have a sufficiently large dataset.

We additionally create an ensemble model to combine the feature sets we have extracted. Because the TF-IDF scores are sparse and the stylometric and linguistic features are dense, we train a base classifier on just the TF-IDF features, and use its predictions as a new dense features set which we combine with the stylometric and linguistic features. We hypothesize that by enriching the feature space of the text corpus, the classification with improve. This stacked model is shown in Figure 6.

2.4 Evaluation

Because there is a bit of a class imbalance from some users sending more emails than others, looking only at the overall accuracy of the model might be misleading. To gain a clear understanding of the performance of our models, we generated confusion matrices to look at accuracy, precision, recall, and F1-scores.

3 Plan & Experiment

3.1 Hypotheses

Following the experiments done in [5], [6], and [7], SVM is the model expected to outperform the rest. However, there is a possibility our ensemble model will outperform the classic models due to the enhanced feature set. To properly compare all the models equally, a classification report is generated after each model classifies the test dataset which allows comparison of the recall, accuracy, F1 score, and precision of the models. The training and test datasets are to remain consistent between the models to ensure that there is no bias between easier/harder data to classify. As a balance of recall and precision, the F1 score is used as the deterministic factor for determining which model is better.

Model	Acccuracy	Precision	Recall	F1
LR	0.85	0.85	0.82	0.83
RF	0.60	**0.93**	0.52	0.57
NB	0.66	0.65	0.66	0.65
SVM	**0.88**	0.88	**0.87**	**0.87**

Table 1: Classification results using TF-IDF features.

Model	Acccuracy	Precision	Recall	F1
LR	0.30	0.3	0.23	0.19
RF	**0.61**	**0.63**	**0.58**	**0.59**
NB	0.11	0.12	0.12	0.06
SVM	0.24	0.18	0.20	0.17

Table 2: Classification results using Stylometric features.

Model	Acccuracy	Precision	Recall	F1
LR	**0.22**	0.2	**0.16**	**0.12**
RF	**0.22**	**0.26**	**0.16**	**0.12**
NB	0.11	0.14	0.14	0.07
SVM	**0.22**	0.15	**0.16**	0.11

Table 3: Classification results using Linguistic features.

3.2 Experimental Design

Here is our overall experimental design:

1. Clean the email textual data
2. Extract feature sets from the textual data (TFIDF, Stylometric, Linguistic)
3. Split the data into train/test
4. Train and evaluate classification models on each feature set alone
5. Train and evaluate stacked model
6. Compare performance

This setup allows us to assess what type of features and models perform best for short-form email classification. It also enables us to tweak or alter at any step to test for consistency and variation in the models' performance. Cleaning and filtering the data differently will yield different results, but isn't the main focus of this project.

4 Results

In all model evaluations, accuracy, precision, recall, and F1 scores were calculated as macro averages across all classes. Tables 1, 2, and 3 show the comparison of the classification models' performances using only the TF-IDF, stylometric, and linguistic features, respectively.

The results of the ensemble model were essentially the same as using only the TF-IDF scores, shown in Table 1. This is because the stylometric and linguistic features we extracted weren't very powerful by themselves, as seen in Tables 2 and 3. We suspect this is due to how short our text documents are. Because the emails are so short, the amount of useful information we can extract is significantly limited. We further hypothesize that the main difference in writing styles of these Enron employees is the difference in their vocabularies. This could include unique email sign-offs or common misspellings.

To further analyze the vocabularies of each user, we looked at the words with the highest TF-IDF scores for two users. These words are relatively more important when attempting to classify their emails. For example, Tables 4 and 5 give the top three words of Vince Kaminski and Matthew Lenhart. The word 'Vince' has a very high TF-IDF score because Vince signs off with this in most of his emails. On the other hand, Matthew's top words don't have as high of a TF-IDF score. Even though he characteristically misspells certain words and uses his name in his emails, it isn't nearly as common as Vince's unique vocabulary. This is reflected in the classification model's ability to easily classify the emails of Vince but not Matthew. Hence, it may be difficult to improve the features from TF-IDF scores as these capture differences in vocabulary extremely well.

Word	TF-IDF
vince	12.85
kaminski	2.26
wicek	2.21

Table 4: Top words for Vince Kaminski.

Word	TF-IDF
definately	2.44
lenhart	2.21
foward	2.11

Table 5: Top words for Matthew Lenhart.

5 Conclusion

By engineering linguistic features and creating a model based on these features, we learn about the the difficulties in developing features that are both indicative of a person's writing and broad enough to cover all emails. We use telltale stylistic linguistic features that indicate to a fellow coworker who authored an email, but since many emails do not contain these unique features, this implementation proves ineffective. It shows that focusing on features that are broadly applicable to all emails would prove more effective than these features and our results show this as well.

There are many things we'd love to try if given more time. Some simple additions would be adding more classification models as well as adding validation sets in order to tune hyperparameters of each model. This would likely increase the performance a bit, as our models weren't tuned.

We also think the ensemble model could be improved upon. In general, the robustness of ensemble models increases with the number of base classifiers. We could extract more feature sets from the text corpus, get predictions of each, and ensemble those predictions into a final prediction (either through majority vote or running it through an additional classifier).

Finally, there are a few ideas we noted while reading through papers that could be applied to the project. Delta TF-IDF is a modification of TF-IDF which has showed improvement in sentiment analysis [9]. We think this could be transferable to authorship attribution. We also think deeper learning models could be applied here due to our dataset being sufficiently large. In particular, CNNs with n-gram features have shown promising results in short-text classification problems like ours [12]. Finally, instead of building an ensemble model to combine features, we could use feature selection methods to identify the most important features for authorship attribution.

6 GitHub Repository Link

Email Classification Repository

7 References

[1] Ibrahim S I Abuhaiba and Mohammad F Eltibi. Intelligent systems and applications. *Intelligent Systems and Applications*, 6:27–39, 2016.

[2] Benedikt Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M. Nickel. Explainable authorship verification in social media via attention-based similarity learning. 10 2019.

[3] William W. Cohen. Enron Email Dataset. https://www.cs.cmu.edu/~enron/, 2015. [Online; accessed 1-December-2022].

[4] Caio Deutsch and Ivandré Paraboni. Authorship attribution using author profiling classifiers. *Natural Language Engineering*, 2022.

[5] Saeed Ul Hassan, Mubashir Imran, Tehreem Iftikhar, Iqra Safder, and Mudassir Shabbir. Deep stylometry and lexical syntactic features based author attribution on plos digital repository. volume 10647 LNCS, pages 119–127. Springer Verlag, 2017.

[6] David I Holmes. Authorship attribution. *Computers and the Humanities*, 28(2):87–106, 1994.

[7] Renkui Hou and Chu Ren Huang. Robust stylometric analysis and author attribution based on tones and rimes. *Natural Language Engineering*, 26:49–71, 1 2020.

[8] Matthew L Jockers and Daniela M Witten. A comparative study of machine learning methods for authorship attribution.

[9] Justin Martineau and Tim Finin. Delta tfidf: An improved feature space for sentiment analysis. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3, pages 258–261, 2009.

[10] Monika Nawrot. Automatic author attribution for short text documents, 2011.

[11] Biveeken Vijayakumar and Muhammad Marwan Muhammad Fuad. A new method to identify short-text authors using combinations of machine learning and natural language processing techniques. *Procedia Computer Science*, 159:428–436, 1 2019.

[12] Haitao Wang, Jie He, Xiaohong Zhang, and Shufen Liu. A short text classification method based on n-gram and cnn. *Chinese Journal of Electronics*, 29(2):248–254, 2020.

Group Member Activities

Sam Smith:

- Pre-processed and cleaned data
- Produced data visualizations
- Set up modeling pipeline

Jackson Ingraham:

- Model Implementation
- Conclusion and Proofreading

Corey Capooci:

- Research and writing for background and introduction.
- Linguistic feature engineering.
- Linguistic model development.
- Linguistic feature analysis.