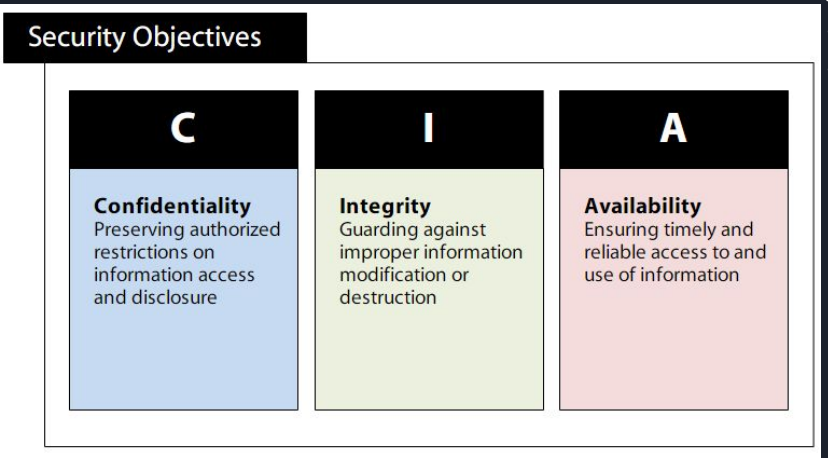
Authorship Identification: Graduate Project

Daniel Kiely
Montclair State University
Master of Science, CyberSecurity
Advisor: Professor C. Leberknight

Presentation Outline

Introduction

- **What is the topic area?**
  - Authorship Identification & Stylometry are the primary areas of focus
  - Purpose was to focus on stylometric features and demonstrate the implementation of extracting features from articles written by 50 different authors
- **Why is this question important to answer?**
  - With regards to the scope of this research, in terms of the Confidentiality, Integrity, and Availability (CIA) triad within the Cyber Security discipline, this research is primarily focused on the integrity aspect of data transmission security.
  - The CIA triad is a fundamental principle that forms the basis for CyberSecurity practices and when all of the segments have been met, the system is considered secure. Ensuring the integrity of data is primarily focused on ensuring the transmission of data was not tampered with. The integrity of data must remain as was originally created by its true curator and if it is not, the data was tampered with and cannot be considered reliable.



Research Problem: What problem does this solve?

Authorship Analysis and Identification

Plagiarism Detection: 'turnitin.com'

- Promote academic integrity by providing feedback to users based on the level of plagiarism found within the document being analyzed between sets of known authors documents.

Decision support systems within a military setting

- If a memo's integrity is compromised, it can severely affect the outcome of a military operation
- Example: If a Supply Chain update memo describing the number of troop supply at a given location and at specific time is modified it can make an operation/mission fail.

TurnItIn: Papers of MSU students vs. Created Model

- TurnItIn provides the ability to detect plagiarism. This project demonstrates how to apply stylometric features to accomplish this task

- On the following slides examples are presented of TurnItIn's plagiarism detection.

Example

Submitted Files: (click to load)

1% Ethics Essay.docx

AI Tools

When one thinks about getting a job, it is usually the same process every single time a it is actually super complicated just to get hired. America as a whole definitely gives the residents a hard time just to get a job. Back in history, there was a point in time when getting involved in the work field required certain things and one that can be mentioned is simply off being a male. AI tools are used more often in jobs today and that would be a more complicate system to be a part of. Comparing our past to the present, we live in a more fair society, where men and women cooperate together when working a job. Women were not permitted to work jobs and had to stay at home but in today's world the AI tools are not biased against them and see no difference between the two genders when it comes to the workforce.

Robot technology

When it comes to acquiring a job, the process is usually the same every time, and getting hired is actually quite difficult. The United States of America as a whole makes it difficult for residents to find work. Back in history, there was a time when entry into the workforce necessitated certain characteristics, one of which might be cited as simply being a man. Today, AI tools are being utilized more frequently in the workplace, making it a more complex system to be a part of. When we look back at our history and compare it to today, we can see that we live in a more equal society, where men and women work together in the workplace. Women used to be barred from working and forced to stay at home, but in today's society, AI tools are not prejudiced against them and see no difference between men and women.

Match Overview

60%

1 Submitted to Montclair... 59%
Student Paper

2 Submitted to North So... 1%
Student Paper

Example (contd.)

Works Cited

Dastin, J. (2018, October 10). Amazon scraps secret AI recruiting tool that showed bias against women. Reuters. Retrieved November 20, 2021, from https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.

Daley, S. (2021, March 31). Women in tech statistics show the industry has a long way to go. Built In. Retrieved November 20, 2021, from https://builtin.com/women-tech/women-in-tech-workplace-statistics.

Holmes, A. (2019, October 8). Ai could be the key to ending discrimination in hiring, but experts warn it can be just as biased as humans. Business Insider. Retrieved November 20, 2021, from https://www.businessinsider.com/ai-hiring-tools-biased-as-humans-experts-warn-2019-10.

Kinkade, R. (2019, March 1). How recruiters can benefit from Automated Text & Calling Systems. Mass Text Messaging & Automated Calling. Retrieved November 20, 2021, from https://www.text-em-all.com/blog/how-recruiters-can-benefit-from-

Works Cited

Dastin, J. (2018, October 10). *Amazon scraps secret AI recruiting tool that showed bias against women*. Reuters. Retrieved November 20, 2021, from https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.

Daley, S. (2021, March 31). *Women in tech statistics show the industry has a long way to go*. Built In. Retrieved November 20, 2021, from https://builtin.com/women-tech/women-in-tech-workplace-statistics.

Holmes, A. (2019, October 8). *Ai could be the key to ending discrimination in hiring, but experts warn it can be just as biased as humans*. Business Insider. Retrieved November 20, 2021, from https://www.businessinsider.com/ai-hiring-tools-biased-as-humans-experts-warn-2019-10.

Kinkade, R. (2019, March 1). *How recruiters can benefit from Automated Text & Calling Systems*. Mass Text Messaging & Automated Calling. Retrieved November 20, 2021, from https://www.text-em-all.com/blog/how-recruiters-can-benefit-from-automated-text-calling-systems.

to find work. Back in history, there was a time when entry into the workforce necessitated certain characteristics, one of which might be cited as simply being a man. Today, AI tools are being utilized more frequently in the workplace, making it a more complex system to be a part of. When we look back at our history and compare it to today, we can see that we live in a more equal society, where men and women work together in the workplace. Women used to be barred from working and forced to stay at

against them and see no difference

We will break down why AI tools

then discuss the various difficulties

decade in this essay. We'll dissect t

employed by various companies ov

along the way. As well as the positi

whether AI-based tools are biased

reasons behind this.

First and foremost, people may wonder why AI technologies were required in the first place. The computer is something that has shown to be quite valuable over time, and it has made everything so much easier. Much easier to collect data and do a variety of other things. Take, for example, technology. "These" were the words of reporter Aaron Holmes, who broke down the use of AI techniques. Tools look through a large number of resumes in a short amount of time

Submitted to Montclair State University on 20...

Student Paper

and one that can be mentioned is simply off of being a male. AI tools are used more often in jobs today and that would be a more complicated system to be a part of. Comparing our past to the present, we live in a more fair society, where men and women cooperate together when working a job. Women were not permitted to work jobs and had to stay at home but in today's world the AI tools are not biased against them and see no difference between the two

60

Match Breakdown

1 Submitted to Montclair... 59
Student Paper

Match 1 of 16

Submitted to Montclair... 59
Student Papers - 2 papers

Submitted to Montclair Stat... 5

Submitted to Montclair Stat... 5

Submitted to Internatio... 9
Student Papers - 3 papers

jolt.richmond.edu 8
Internet Source

Submitted to California... 7
Student Paper

Exclude Sources

Code Stylometry

What is code stylometry?

- Stylometry vs. Code Stylometry
    - Stylometry: Netherlands Institute for War Documentation of Anne Frank's Diary

Caliskan-Islam, A., Harang, R., Liu, A., Narayanan, A., Voss, C., Yamaguchi, F., & Greenstadt, R. (2015). De-anonymizing programmers via code stylometry. In *24th USENIX security symposium (USENIX Security 15)* (pp. 255-270).

- Code Stylometry: Extracting features of a code the same way stylometry works. It is frequently used identify malware as was seen with Sony Pictures Entertainment in 2014

Code Stylometry Example

- The source of the 2014 Sony Pictures Entertainment hack, in which the group leaked personal data of employees within the company, was found to be North Korea. The conclusion was made after experts found similarities between the code used during this attack and the malware Shamoon
- Shamoon erased the computers infrastructure.
- “A technical analysis of the data deletion malware used in this attack revealed links to other malware that the FBI knows North Korea previously developed…there were similarities in specific lines of code.”-F.B.I

Literature Review

1. R. Ramezani work on n-grams across multiple languages including an English dataset as well as a Persian dataset. His work fell short in that he simply looked one document compared to another singular document.
    a. Used of TF_IDF algorithm for extracting features from the given work
    b. Inspired the use of N-grams, as it was one of the more successful features used.

2. Fuchen Peng, Dale Schuurmans, Vlado Keselj, and Shoajun Wangs' work was focused primarily on creating a language independent model.
    a. fell short in that it did not have texts of uniform length
        i. difficult to find uniform features in shorter text documents
    b. only comparing one document to another and not multiple text documents of various authors.

Challenge of Authorship Identification

- Typical authorship identification tests to see if an article was written by a specific author.
- Where this task varies from typical methods is that our task involved checking an article against many authors.
- Seen in previous Literature Review Section

Research Question

- “The purpose of this paper is to conduct an experiment in which stylometric features can be extracted and further analyzed to accurately predict the author of a document when comparing them to a set of known authors within a test data set” (Kiely, 1).

- Integrity of Data is of the utmost importance and this can verify the integrity of data when comparing a document or piece of malicious code to an original document or source code that was previously implemented.

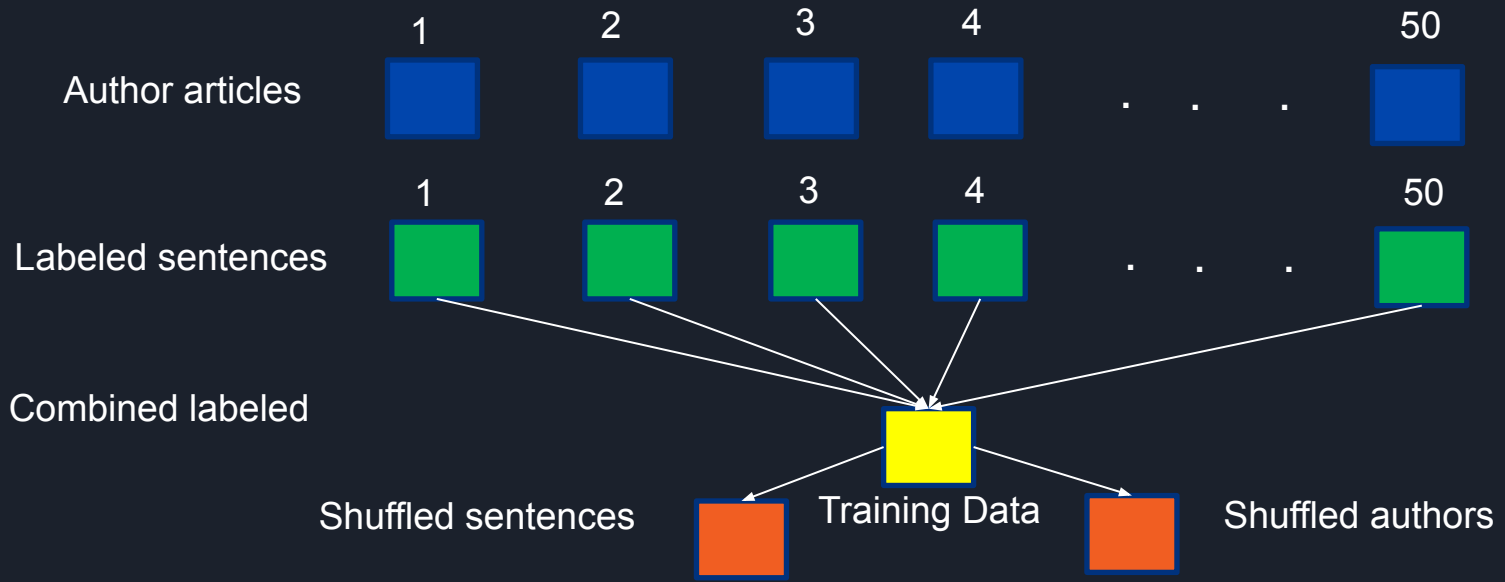
Dataset

- Based on articles written by 50 authors.
- 50 authors
- 50 articles per author totaling 2,500 text documents
- Average length of article was 250 words
- Originating curation: National Engineering Research Center for E-learning in Wuhan China
    - Made available for downloadable purposes by University of California-Irvine: Machine Learning Repository.

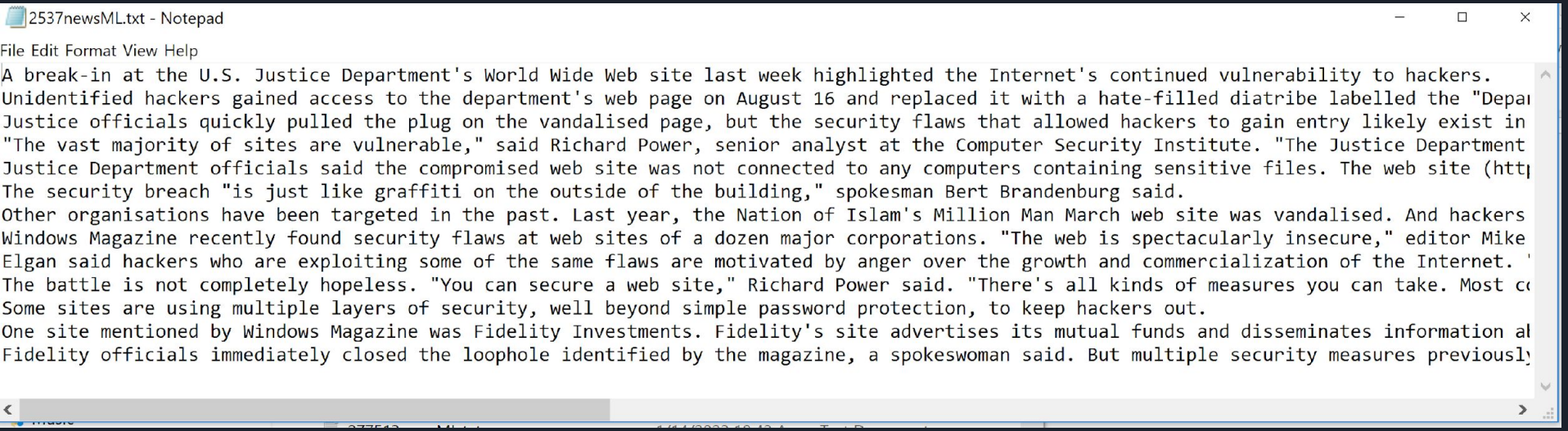
Data Pre-processing

Data-Preprocessing Steps

1. Label each sentence for each author
2. Combine all labeled author data sets into 1 file
3. Shuffle the combined file
4. Split the combined file into a combined shuffled sentences and shuffled authors



Sample article



List of authors

```
# Create target names (author names) or categories for each article
# These target names or categories will be used by both the training and test datasets
from IPython.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))

import os
path=os.path.dirname("/Users/DanKiely/Documents/C50/C50train/")
#print(path)
#print(os.path.basename(path))
authors = []
rootdir = path
for file in os.listdir(rootdir):
    d = os.path.join(rootdir, file)
    if os.path.isdir(d):
        authors.append(os.path.basename(d))
#        print(os.path.basename(d))
#        print(d)
print(authors)
```

```
['RobinSidel', 'LynnleyBrowning', 'KouroshKarimkhany', 'MichaelConnor', 'JoeOrtiz', 'EricAuchard', 'AaronPressman', 'SimonCowell', "LynneO'Donnell", 'EdnaFernandes', 'Kev
evinDrawbaugh', 'KarlPenhaul', 'MartinWolk', 'ScottHillis', 'DavidLawder', 'FumikoFujisaki', 'MarcelMichelson', 'NickLouth', 'DarrenSchuettler', 'WilliamKazer', 'TanEeLyn
t', 'BradDorfman', 'AlanCrosby', 'JonathanBirt', 'BenjaminKangLim', 'TheresePoletti', 'KeithWeir', 'JoWinterbottom', 'MarkBendeich', 'JaneMacartney', 'MatthewBunce', 'Tod
nardHickey', 'KirstinRidley', 'AlexanderSmith', 'LydiaZajc']
```

Create training data

```
import glob
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

for author in authors:
    read_files = glob.glob("/Users/DanKiely/Documents/C50/C50train/"+ author + '*.txt')
    with open("/Users/DanKiely/Documents/C50/C50trainCombined/"+ author + '_AllData.txt', 'wb') as outfile:
        for f in read_files:
            with open(f, 'rb') as infile:
                outfile.write(infile.read())

for author in authors:
    with open("/Users/DanKiely/Documents/C50/C50trainCombined/"+ author + '_AllData.txt', 'r') as f:     # load file
        lines = f.read().splitlines()  # read lines

    with open("/Users/DanKiely/Documents/C50/C50trainLabeled/"+ author + '_labels.txt', 'w') as f:
        f.write('\n'.join([line + '\t' + author for line in lines]))  # write lines with author name appended
```

This function creates the training files with labels

Create training data (contd.)

```
import glob
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

read_files = glob.glob("/Users/DanKiely/Documents/C50/C50trainLabeled/*.txt")
with open("/Users/DanKiely/Documents/C50/C50trainFinal/Trained_Data_Labels.txt", 'wb') as outfile:
    for f in read_files:
        with open(f, 'rb') as infile:
            outfile.write(infile.read())
```

Sample of training data

In [2]:

```
# Display the train labeled data

import glob
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

df_train_with_labels= pd.read_csv("/Users/DanKiely/Documents/C50/C50trainFinal/Trained_Data_Labels.txt", sep = '/n', engine ='python', header=None)
#df['author_id'] = df[0].factorize()[0]
#df['author_id'] = 0
#df.head(20)
df_train_with_labels
```

Out[2]: 'The Internet may be overflowing with new technology but crime in cyberspace is still of the old-fashioned variety.\tAaronPressman'

- Out[2] shows the label “AaronPressman” tying that line to the specific Author. This was done for each line of each author

Test data

The same pre-processing steps were performed for the test data

1. Label each sentence for each author
2. Combine all labels author data sets into 1 file
3. Shuffle the combined file
4. Split the combined file into a combined shuffled sentences and shuffled authors

```
# Combine the test data and labels into 1 dataframe so we can shuffle and extract the separate label and test data while
# and maintain the correct labels for each test data
import pandas as pd

df_test_data['author_name'] = df_test_labels['author_name']

################################## Shuffle test_data ##########################
test_data_df = df_test_data.sample(frac = 1)
test_data_df
################################## Extract the label and test data into 2 separate dataframes ###############
test_data = test_data_df['article']
test_labels = test_data_df['author_name']
test_data = test_data.tolist()
test_data
```

Predict labels for test data

```
# https://www.youtube.com/watch?v=60pqgfT5tZM

%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import pandas.util.testing as tm


from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline

train_data = np.array(train_data_articles)
train_labels = np.array(train_data_authors)
#test_data = np.array(test_data)

#############################################################################################################
#############################################################################################################
# Creating a model based on the MultiNomial Bayes
#model = make_pipeline(TfidfVectorizer(), MultinomialNB())  # without stop words
#model = make_pipeline(TfidfVectorizer(stop_words='english'), MultinomialNB())  # with stop words
model = make_pipeline(TfidfVectorizer(ngram_range=(1,5), stop_words='english'), MultinomialNB())

model.fit(train_data, train_labels)

# Creating labels for the test data
labels = model.predict(test_data)
print(labels)
```

Features & Multinomial Bayes

- ngram_range(1,5) is a parameter that sets the lower and upper boundary of the range of n-values to be extracted.
  - in our case we used up to 5
  - ngram_range(1,1) would mean only unigrams are to be extracted.
  - ngram_range(2,2) means only bigrams are to be extracted
  - ngram_range(1,3) means that unigrams, bigrams and trigrams are to be extracted.

unigram example:

- [Authorship] [Analysis] [is] [very] [Interesting] [and] [should] [be] [widely] [used].

n-gram (5,5) example:

- [Authorship Analysis is very interesting] [and should be widely used].

- TF-IDF
  - algorithm used to weigh a keyword/phrase in a text document and assign weight/importance to that keyword based on number of times that is shown in the document(s).
  - TF looks at a specific document for frequency of a particular keyword with relativity to the specific document
  - IDF focuses on how common/uncommon a words is against the corpus.

Multinomial Bayes

Multinomial Naive Bayes Classification is commonly used for assigning documents to classes based on statistical analysis of the contents Multinomial Naive Bayes works by a giving a term a numerical representation based on how many times it appears (frequency)

- Predicts the probability of each tag of text (Authors Name) and then classify it with the highest probability tag.
- Multinomial Naive Bayes is suitable for the analysis and classification of ‘discrete features’
    - word counts (uni-grams)
    - frequency of words (TF)
    - term length (n-grams)

Results

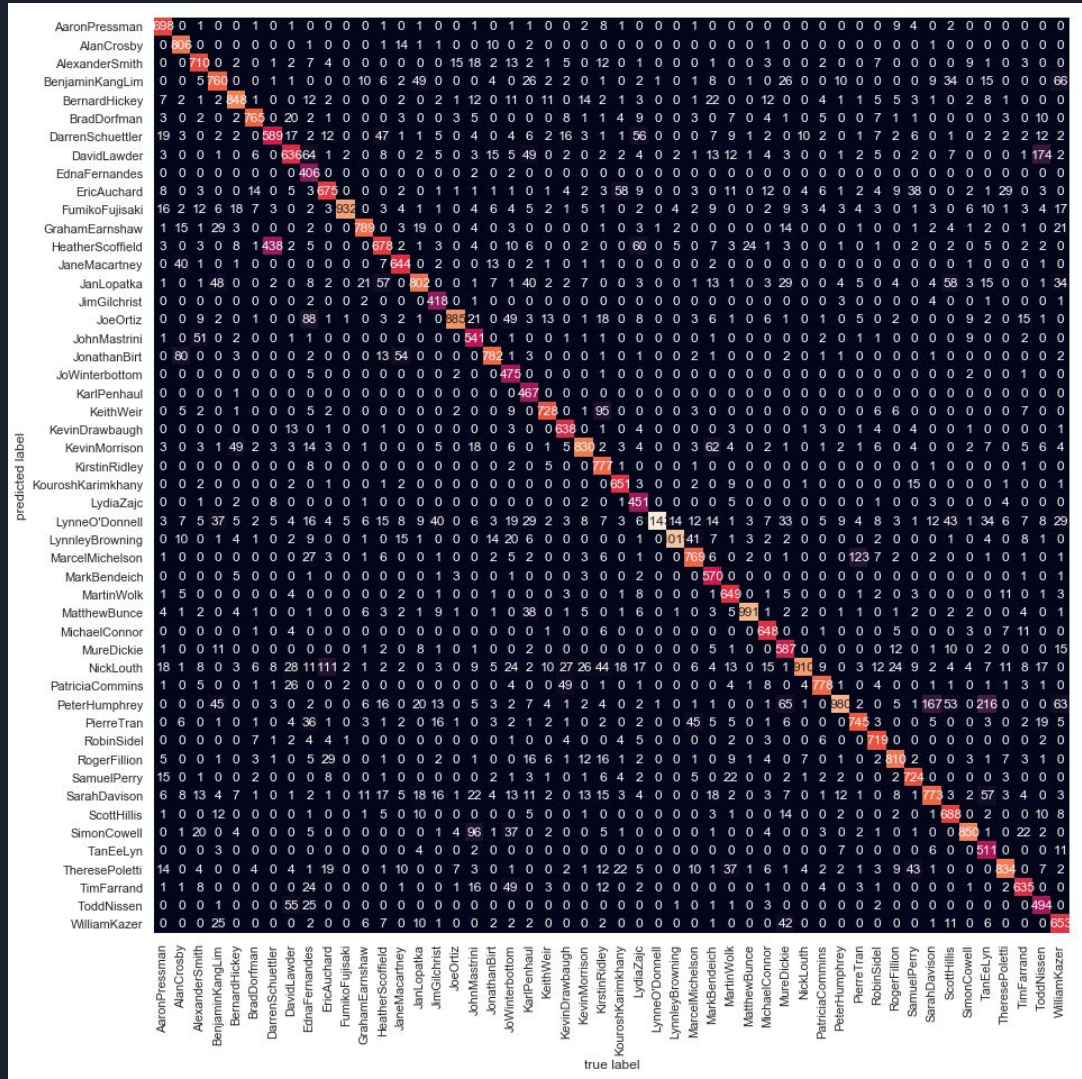
Experiment

- Test data consists of articles from all 50 authors
- training data consists of data from all 50 authors
- Comparing All documents to all potential authors and seeing how accurate our features can identify the correct placement for each author

Experiment

- Test data consists of articles from all 50 authors and training data consists of data from all 50 authors

```
Recall:     0.8074981382656972
Precsion:   0.8406148807138711
F1:         0.8237237900046066
```



Limitations

- Models like ours tend to do better with larger amounts of data
  - we did not look at what happens with smaller amounts of data such as only a 100 files total.
- we were attempting to validate previous research and recreate their results based on the features they had selected.
  - Would be interesting to see if the model held its levels of precision while look at sentence length, verb/adjective usage, or other broad features.
-

CONCLUSION & FUTURE WORK

Although the results were not as accurate as other models used by other researchers, this experiment allowed us to extract features and with a precision rate of 84% for the second experiment. The first experiments nearly 50% precision rate gave us insight into future work to look into:

- Different features instead of unigrams and n-grams. Potentially positive and negatively loaded words, adjective usage, sentence structure/length.
- If this model would hold up with different programming languages for code stylometry

This allowed us to successfully extract unigrams - *5-grams* in order to attempt to identify an author based on the the terms frequency compared to the entire corpus. This experiment allowed us to further the research and experiments in the realm of authorship analysis. Turnitin limitations are the data sample size. It is only comparing