AUTHORSHIP IDENTIFICATION OF ONLINE MESSAGES: A COMPARATIVE STUDY

Team Members: Chetan Manjesh, Rakshitha K Bhat

ABSTRACT

Authorship Identification determines the likelihood of a piece of writing to be produced by a particular author by examining other writings by that author. It is a classification problem whose complexity level can be determined by several parameters such as the kind of feature set used, the size of the training data, the number of authors considered, the number of writings per author, the type of classification model used etc. This project was developed to perform the task of classifying authors of online messages taken from Reddit and Enron Email Dataset. A 2-way SVM classifier was developed which achieved an accuracy of 83.5% on the Enron Email Dataset and an accuracy of 74% on the Reddit Dataset. The classification parameters were then altered to compare the effect of these parameters on classification accuracies.

INTRODUCTION

In today's world due to the freedom given to users to create unlimited identities online, malicious entities have taken advantage of this to become untraceable and use that as an opportunity to carry out unethical and criminal activities. Online messages are often used by criminals as a platform for distributing illegal items such as drugs, child pornography and also for making threats and spreading messages that often disrupt social harmony etc. that are hard to pinpoint to one source as a result of methods of anonymity adopted such as using multiple usernames and masked IP addresses. Due to immense volume of such messages there is an imperative need for automated methods for detecting such users. This project seeks to use natural language processing models for going past such deception and identifying the authors of online messages using supervised learning. The models are trained using the features extracted from the online messages generated by the users and then classifying new messages to the author that has most likely written them.

In essence, authorship identification is a classification problem. The complexity level of this problem can be determined by several parameters such as the kind of feature set used, the size of the training data, the number of authors considered, the number of writings per author, the type of classification model used etcetera. The accuracies differ when these parameters are changed. The project hopes to answer this problem by

recognising the effect of various parameter changes to the accuracy of the classifier and choosing those parameters that have an optimal effect on classifier accuracy. We changed the number of training messages per author and noticed an increase in accuracy with increase in the training set. Similarly on increasing the number of authors to be identified, the accuracies seem to decrease. The classifier was able to classify the messages of 2 authors much better than when it was subjected to classify among 10 authors. We also studied the effect of features on classification of authors. It was seen that lexical features and structural features helped to differentiate the authors much better than the syntactic features. It then performed a comparative study to assess the accuracy and potency of each classification model. It was observed that SVM seem to be better suited for this task than Naive Bayes classifier.

RELATED WORK

[1] A framework for authorship Identification of Online Messages: writing-Style features and classification Techniques (Zheng et al., 2006):

We are basing our project off the work done in this paper. This paper talks about developing a feature set using various style features and examining the prediction power of authorship-identification method for online messages under different parameter settings. Some of the common characteristics of online messages as mentioned in the paper are:

- Limited length of online messages
- The structural style used in online messages is often different from normal text documents. Hence using special features like structural layout traits may be useful in forming a suitable feature collection.
- Online messages are multilingual in nature.
- The number of potential authors for an online message could be large.

Keeping these characteristics in mind, the authors propose to extract 4 categories of features from the messages namely Lexical Features, Syntactic Features, Structural Features and Context Specific Features. These features were then used to develop classifier models which was then used to predict the author of an unknown message. Three different classifiers : C4.5 decision tree (Quinlan, 1986), backpropagation neural networks (Lippmann, 1987), and SVM (Cristianini & Shawe-Taylor, 2000) were used to compare the accuracies of different classification techniques. For the purpose of our project, we restricted ourselves to implementing three categories of features namely

Lexical, Syntactic and Structural features and implementing 2 classification models namely Naive Bayes and Support Vector Machines.

[2] Authorship Attribution with Support Vector Machines (J. Diederich et al., 2003):

This paper investigates the efficacy of Support Vector Machine models and compares them with other models commonly used for authorship attribution such as Naive Bayes model and Decision trees and concludes that it is much better than these models. The authors of the paper discover that this model is robust and works remarkably well when it has to identify text pieces of the author that are different from the topic it was trained on. They also report that classifier accuracy does not decrease with an increase in the number of features after a point as with the previous models and as such there is no need for feature subset selection or feature preprocessing. The paper discovers that the training time is nearly the same as the taken to train neural network models and that bigram uses of functional words have a negative impact on classifier accuracy. Hence in our implementation, we have used functional words in their unigram form and have decided to implement a Support Vector Machine for author identification.

[3] Computational Constancy Measures of Texts—Yule's K and R´enyi's Entropy, Computational Linguistics(K. Tanaka-Ishii, S. Aihara)

This paper talks constancy measures for the works of a particular author such as Yule's K measure, Simpson's D measure and Hapax Legomena . These measures are representative of the writing style of the author from the point of view of vocabulary richness and have been implemented in our project based on the formulae given in the paper. It is mentioned that these measures do not change for works of a different works of specific authors provided the texts are above a certain size i.e. these values converge after a particular file size and also measures how constancy varies across different texts. It also explores the relationship between different constancy measures and investigates the extent to which they are useful for distinguishing between different authors. Since these measures work effectively only for messages above certain thresholds the messages we have extracted also need to meet a certain threshold for the purpose of classification.

[4] Mining E-mail Authorship (Olivier de Vel)

This paper tells us that structural features have a strong impact on classifier accuracy. The paper also says that categorization performs better for some authors than others implying

that there is a need for greater number of features to be identified. Hence the use of structural features increases the accuracy of the classifier.

[5] Another Perspective on Vocabulary Richness(D.L. Hoover)

This paper states that vocabulary richness measures used above are not very useful when there is wide variation or diversity among the texts(intertextual variation) or even within a particular text(intertextual variation) churned out by a particular author. This paper also states that vocabulary richness methods perform worse for distinguishing between different authors when the number of textual documents supplied for a particular author are more and hence demonstrates that there is need for developing other measures of vocabulary richness. This helps us identify possible flaws in using vocabulary richness measures as features and how these set of features could affect the overall accuracy of the model.

[6] Computer-based authorship attribution without lexical measures.(Stamatatos et al., 2001):

This paper talks about the improved efficacy of token-level, phrase-level, and analysis-level measures over lexical features and vocabulary richness measures. Among the three kinds of features token-level measures such as work counts and sentence boundaries have a better impact on classifier accuracy than phrase level features such as counts of Verb Phrases, Noun Phrases etc. and Analysis-level features. The authors have also discovered these stylometric features with vocabulary richness measures produces an even better accuracy. In our model we have combined stylometric features such as word counts along with vocabulary richness measures.

[7] On the feasibility of internet-scale author identification.(Narayanan, Arvind, et al.)

This paper talks about the efficacy of stylometric features such as lexical features used in our implementation in classifying blog articles that have been trained on a corpus of blog articles belonging to more than one hundred thousand authors. The authors have found that it scales well with an increase in the number of training sets and also that it works well when tested on cross-context blog articles of the same author.

[8] Authorship Identification on Twitter (Antonio Castro et al., 2012):

This paper talks about performing author identification on tweets. The authors of this paper were inspired by [7] and applied the stylometric features used by them for large text pieces to identify whether a high classification accuracy would hold even when they

are used for classifying much smaller tweets. The features used were character based features such as character frequencies and syntactic features such as the frequency of certain functional words for their models. The authors found them to be highly discriminative for the purpose of author identification even for tweets. Since tweets are short textual pieces of length less than 140 characters and since we are performing author identification on online messages/comments which are of short length too, we have extracted similar features for the purpose of the project. This paper also helps to explain why some of the structural and syntactic features fails to increase accuracy for Reddit dataset.

[9] Whose Book is it Anyway? Using Machine Learning to Identify the Author of Unknown Texts(Sean Stanko et al., 2012):

The authors of this paper implemented a multinomial classifier using Support Vector Machines to identify the authors of unknown texts. The performance of the Support Vector Machine Model was found to be better than other models typically used such as N-grams and the output was reduced from the order of minutes to about few seconds without any reduction in classifier accuracy. They also found that classifier accuracy increases with the size of the text. This has been observed in our implementation where the classifier did a better job at classification when there were more training examples provided to it.

[10] N-gram feature selection for authorship identification(John Houvardas et al., 2006):

This paper describes another approach to author identification by using a different set of features namely Character n-grams for identifying the authors. Character n-grams are able to capture nuances in lexical, syntactical, and structural level. Tokenization is not needed when extracting character n-grams, thus making the approach language independent. On the other hand, they considerably increase the dimensionality of the problem in comparison to word-based approaches. Due to this, n-grams of fixed length are generally used. The proposed approach was found to be at least as effective as information gain for selecting the most significant n-grams.

DATA

We are using the following datasets for the purpose of this project:

- Enron Email Dataset
- Reddit Dataset

i) *ENRON EMAIL DATASET*:

The Enron Email Dataset is a publicly available dataset of emails circulated among the senior management of Enron Corporation. It can be downloaded from https://www.cs.cmu.edu/~./enron/ . The size of the decompressed dataset is approx 1.4 GB.

It is organized into folders where there are individual folders for each of the 150 users and has about 0.5 M messages. Each user may contain further subdirectories like *'inbox', '_sent_mail', 'all documents', 'discussion threads'* etc. For the purpose of this project, we are only interested in the mails sent by the individual user. Hence we will be concentrating on obtaining information from the ***'_sent_mail'*** folder. Not all users have sent out emails. After running preliminary analysis on data, we found out that **78 users(52%)** have sent out one or more emails and the remaining 72 have not sent any.

A sample email sent by an employee, allen-p is shown below:

Message-ID: <25123422.1075855686459.JavaMail.evans@thyme>
Date: Wed, 6 Dec 2000 04:43:00 -0800 (PST)
From: phillip.allen@enron.com
To: andrea.richards@enron.com
Subject: Re: Associates & Analysts Eligible for Promotion
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: Andrea Richards
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Dec2000\Notes Folders\'sent mail
X-Origin: Allen-P
X-FileName: pallen.nsf

I would support Matt Lenhart's promotion to the next level.

I would oppose Ken Shulklapper's promotion.

The text in **bold** above is the content of the email. This is the part we are concerned about. Message-ID, Date, From, To, Subject etc are the header fields. These provide more information about the email. Since we are just concerned with identifying authors based on the content written, these fields are removed from the training/test data during the preprocessing step and only the content is supplied to the feature extraction script to extract the features.

From among the 78 users who have sent mails, the top 10 users who have sent the maximum amount of relevant email content are as follows:

User	**Sent Email count**
mann-k	2594
kaminski-v	1657
symes-k	1148
germany-c	1089
bass-e	1081
scott-s	762
rogers-b	742
beck-s	729
arnold-j	668
rodrique-r	647

Table 1: Top 10 authors and their mail count for Enron Dataset

The following table provides details about the average number of words and sentences of each of these authors.

Author	**Average number of words**	**Average number of Sentences**
mann-k	122	10

kaminski-v	139	13
symes-k	117	7
germany-c	87	7
bass-e	111	10
scott-s	153	11
rogers-b	40	3
beck-s	180	10
arnold-j	120	10
rodrique-r	67	6

Table 2: The top 10 authors and their average word length and sentence length for Enron Dataset

The word count and sentence count metrics of the emails sent by the top 10 authors are as follows:

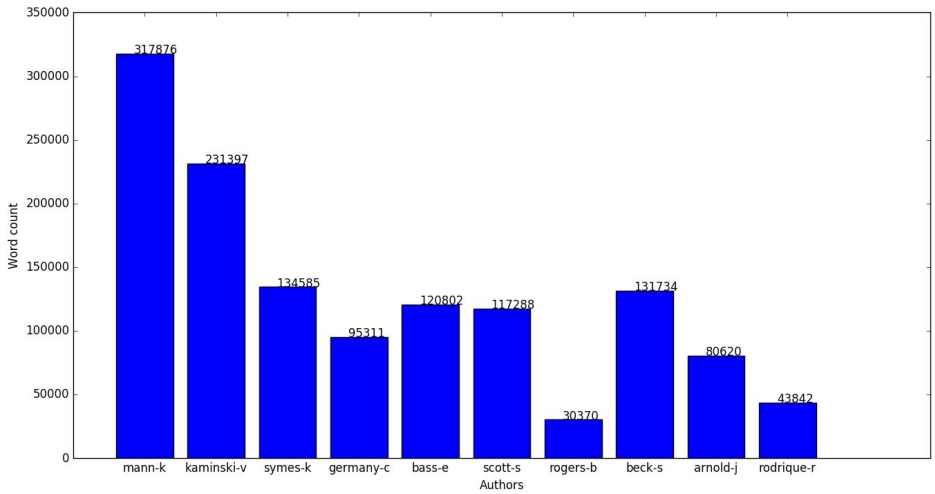

Figure 1: Total number of words written per user

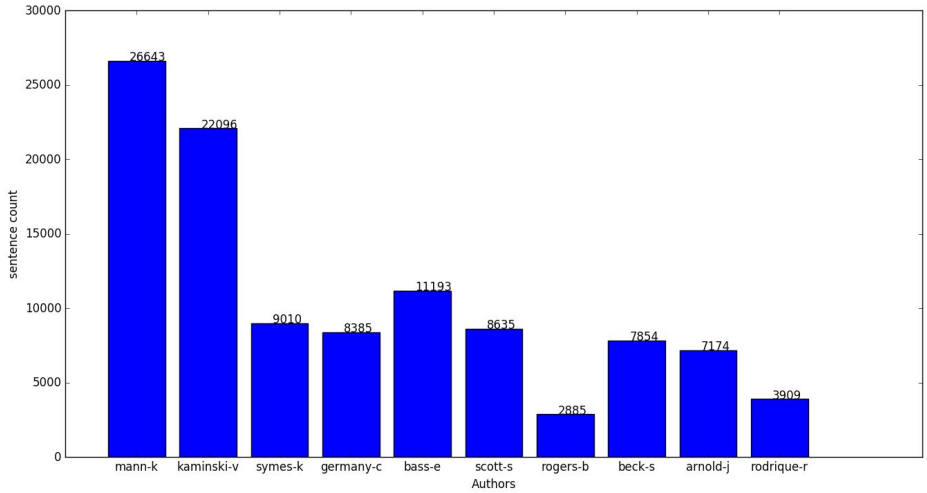

Figure 2: Total number of sentences written per author

This dataset did not require any annotation. Each author had a separate folder containing emails sent by him/her. The dataset was already annotated. Thus no manual annotation was required.

ii) *REDDIT DATA SET* :

We have an entire month's worth of Reddit Comments from January 2015. It can be downloaded from https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/. The size of the uncompressed file was approximately 35 GB.

Each comment is stored as a JSON block and different blocks are separated by newline characters. Each JSON block contains information about the subreddit under which the comment was posted, name of the commenter and various other fields. From this data we considered the the first 3 million blocks and extracted information such as the name of the commenter, the body of the comment and subreddit id under which that comment was posted.

A sample JSON block representing a comment from which information was extracted is as follows:

{"gilded":0,"author_flair_text":"Male","author_flair_css_class":"male","retrieved_on":1425124228,"ups":3,"subreddit_id":"t5_2s30g","edited":false,"controversiality":0,"parent_id":"t1_cnapn0k","subreddit":"AskMen","body":"I can't agree with passing the blame, but I'm glad to hear it's at least helping you with the anxiety. I went the other direction and

started taking responsibility for everything. I had to realize that people make mistakes including myself and it's gonna be alright. I don't have to be shackled to my mistakes and I don't have to be afraid of making them. ","created_utc":"1420070668","downs":0,"score":3,"author":"TheDukeofEtown","archived":false,"distinguished":null,"id":"cnasd6x","score_hidden":false,"name":"t1_cnasd6x","link_id":"t3_2qyhmp"}

From the above data the attributes of the keys relevant to us that were extracted are: "subreddit_id", "subreddit", "body" and "author" along with their corresponding values. It was stored in the format :

{

subreddit_id1:{author_name1:{ “comments” : [ ] , “num_comments” : X }, author_name2:{..},.....},

subreddit_id2:{author_name1:{ “comments” : [ ] , “num_comments” : X } author_name2:{....},.....},

....

}

The aim is to build a classifier for identifying different authors in a specific subreddit. We did an analysis to find those subreddits which have a minimum of 5 authors with more than 50 comments. The purpose of this was to find a subreddit which has substantial data for training and testing.

The plot below contains information about the number of authors with substantial number of comments found found for top 10 subreddits.

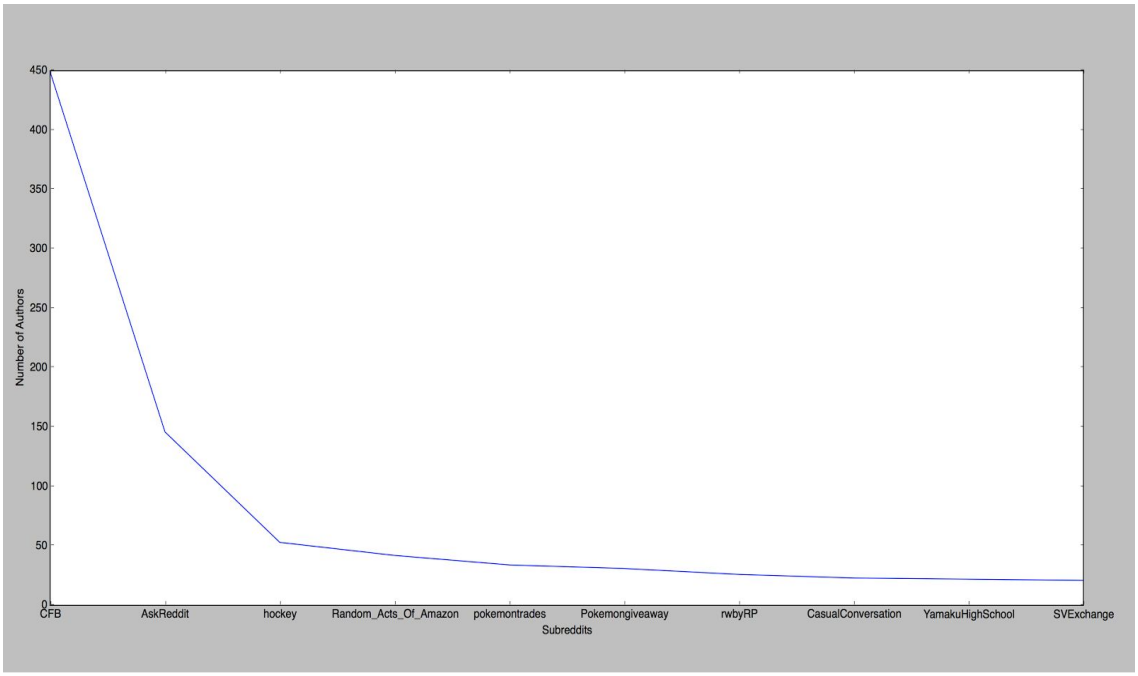

Figure 3: Number of authors per subreddit

The subreddit chosen for building a classifier for its different comment authors is the subreddit /r/CFB because it has the most number of authors who meet the minimum comment threshold of 50-60 comments required for classifying data.

Average number of words per comment among the top 10 most frequently commenting reddit users in /r/CFB:

Reddit user in /r/CFB	**Avg. number of words per comment**
nittanylionstorm07	9
HardKnockRiffe	16
OnthefarWind	13
AJinxyCat	9
740Buckeye	10
GoldandBlue	14
beepbeepimajeep05	10

xDalexx	8
Domthecreator14	8
avboden	8

Table 3: Top 10 commenters and their average words per comment.

The plot below contains information about the number of comments per author for the top 10 commenters of the /r/CFB.

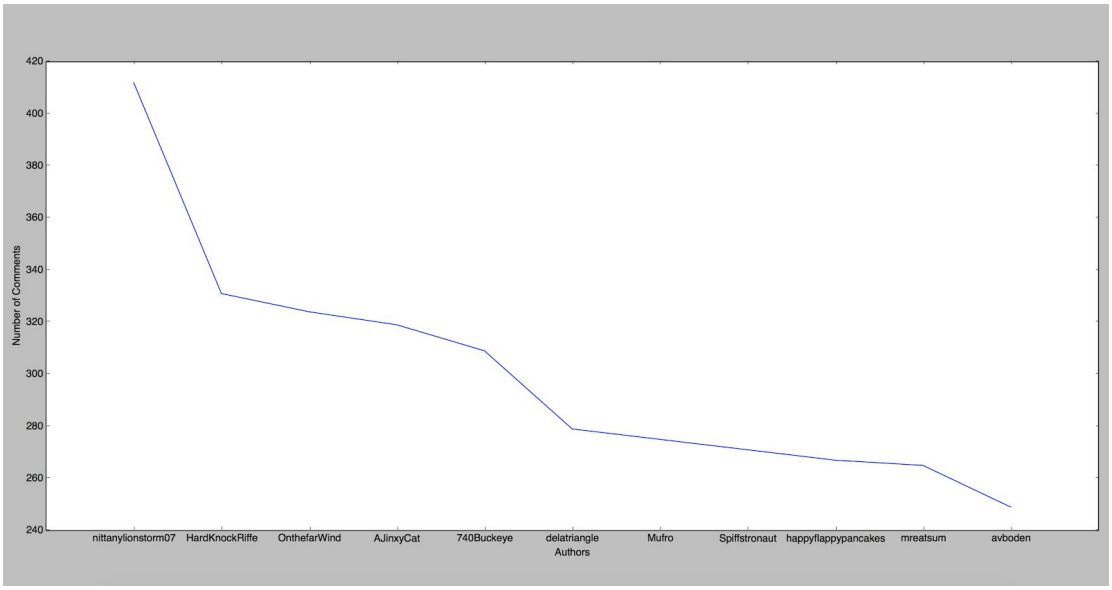

Figure 4: Authors versus comment count for Reddit Dataset

This dataset too did not need any manual annotation because each comment was stored along with the user_id of the reddit commenter who wrote the comment.

METHODOLOGY

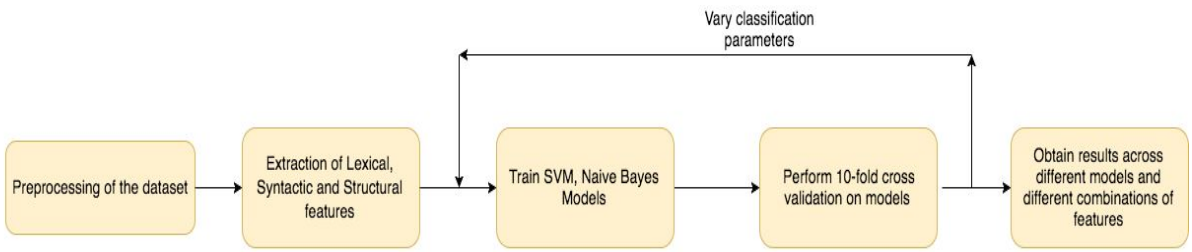

Figure 5: The workflow of the project

i) Preprocessing of the dataset:

The first step is to preprocess the dataset. Both the datasets are preprocessed to remove any unnecessary metadata and get only the content of the users as mentioned in the above section. *preprocessing_enron.py* and *Redit_data_Preprocessing.py* are the two scripts which perform the preprocessing.

ii) Extraction of features:

Three different writing style features were identified for distinguishing between different types of authors in [1] :

1) Lexical Features(F1): Lexical features can be further subdivided into character-based and word-based features.
    a) Character-based features: These features are about the frequency of occurrence of various characters such as digits, alphabets, upper case characters, white spaces, tabs etc.
    b) Word-based features: These features pertain to the measures about the usage of words such as the total number of words, the number of short words, the average number of characters per word and also other measures pertaining to vocabulary richness such as Yule's K measure, Hapax Legomena etc.
2) Syntactic Features(F2): They are used to capture the author's writing style at a sentence level.
3) Structural Features(F3): These features represent the way the author organises the layout of his comments. These features are especially useful and evident in ENRON email dataset where structural features such as salutations and greetings can be particularly useful in distinguishing between two authors.

While the Structural features play an important role in distinguishing between authors for the Enron Email dataset, these features don't seem to add weightage to the Reddit Dataset. This is because Reddit comments, unlike emails, are very short and wouldn't have paragraphs or salutations or greetings. Hence we see a decrease in accuracy for the Reddit dataset when these features are also included. *feature_extraction.py* script takes the content and extract the respective features and returns the feature vectors. The table below lists all the features which were implemented in this project which were mainly taken from the ones mentioned in paper [1].

1) Lexical Features	Character-based features 1. Total number of characters(C) 2. Alphabet distribution ratio:Total number of alphabetic characters/C 3. Uppercase distribution ratio: Total number of uppercase characters/C 4. Digit distribution ratio: Total number of digit characters/C 5. White space distribution ratio: Total number of white-space characters/C 6–31. Frequency of letters (26 features) A–Z 32–53. Frequency of special characters (21 features) ~ , @, #, $, %, ^, &, *, -, _, = ,+, >, <, [,], {, }, /, \, | Word-based Features 54. Total number of words (M) 55. Total number of short words (less than four characters)/M 56. Total number of characters in words/C 57. Average word length 58. Average sentence length in terms of character 59. Average sentence length in terms of word 60. Total different words/M 61. Hapax legomena 62. Hapax dislegomena 63. Yule's K measure 64. Simpson's D measure 65. Sichel's S measure 66. Brunet's W measure 67. Honore's R measure 68–87. Word length frequency distribution /M (20 features)
2) Syntactic Features	88–106 Frequency of punctuations (8 features) ",", ".", "?", "!", ":", ";", " ' ", " " " 107–257 Frequency of function words (150 features): a, between, in, nor, some, upon, about, both, including, nothing, somebody, us, above, but, inside, of, someone, used, after, by, into, off, something, via, all, can, is, on, such, we, although, cos, it, once, than, what, am, do, its, one, that, whatever, among, down, latter, onto, the, when, an, each, less, opposite, their, where, and, either, like, or, them, whether, another, enough, little, our, these, which, any, every, lots, outside, they, while, anybody, everybody, many, over, this, who,

	anyone, everyone, me, own, those, whoever, anything, everything, more, past, though, whom, are, few, most, per, through, whose, around, following, much, plenty, till, will, as, for, must, plus, to, with, at, from, my, regarding, toward, within, be, have, near, same, towards, without, because, he, need, several, under, worth, before, her, neither, she, unless, would, behind, him, no, should, unlike, yes, below, i, nobody, since, until, you, beside, if, none, so, up, your
3) Structural Features	258. Total number of lines 259. Total number of sentences 260. Total number of paragraphs 261. Number of sentences per paragraph 262. Number of characters per paragraph 263. Number of words per paragraph 264. Has a greeting 265. Has separators between paragraphs 254. Has quoted content 266. Position of quoted content 267. Indentation of paragraph 268. Use of e-mail, telephone, url as signature

Table 4: Feature Set implemented in the project

iii) Classification:

The feature vectors generated for the content are then fed to the SVM and Naive Bayes model to train the classifier. We use Scikit to implement a Multinomial Naive Bayes classifier and Support Vector Machine with linear kernel. A 10-fold cross validation was carried out to train and test the models and the average accuracy over the 10 fold was recorded. In order to study the effect of change in parameters to the accuracies, the following variations were performed on the Enron Email Dataset and the results were recorded.

a) We first perform a binary classification by taking the top 2 authors who have sent out the maximum number of emails. We compare the classification accuracy obtained by Naive Bayes and SVM classifiers. The size of the training data is set to 100 emails and the all the features are used for classification.

b) We next vary the feature parameter i.e compare the accuracies obtained by just using lexical features, lexical + syntactic features, lexical + syntactic + structural features. All the other parameters remain the same as mentioned above.

c) Next we change the size of the training set i.e increase the size from 25 emails, 50 emails, 75 emails to 100 emails and compare the accuracies obtained. Other parameters remain the same as mentioned in (a).

d)Lastly to compare the effect on accuracies on increase in the number of authors, we increase the size of authors from 2 authors, 5 authors, 10 authors thereby performing a 2-way, 5-way, and 10-way classification respectively. The top n(where n being 2,4,10) who sent maximum mails are taken into consideration.

RESULTS

The experiments mentioned in the previous paragraph were run on the Enron Email dataset and the following results were achieved.

a) Comparison of classification Models:

Accuracy obtained with Naive Bayes: 69%

Accuracy obtained with SVM: 83.5 %

The SVM model performed better than the Naive Bayes model as expected. Due to the presence of a large number of features(268), the SVM was more capable at handling these many features than Naive Bayes. The Reddit dataset also gave similar results with accuracy of 64% and 74% reported for Naive Bayes and SVM models respectively.

b) Comparison of feature sets:

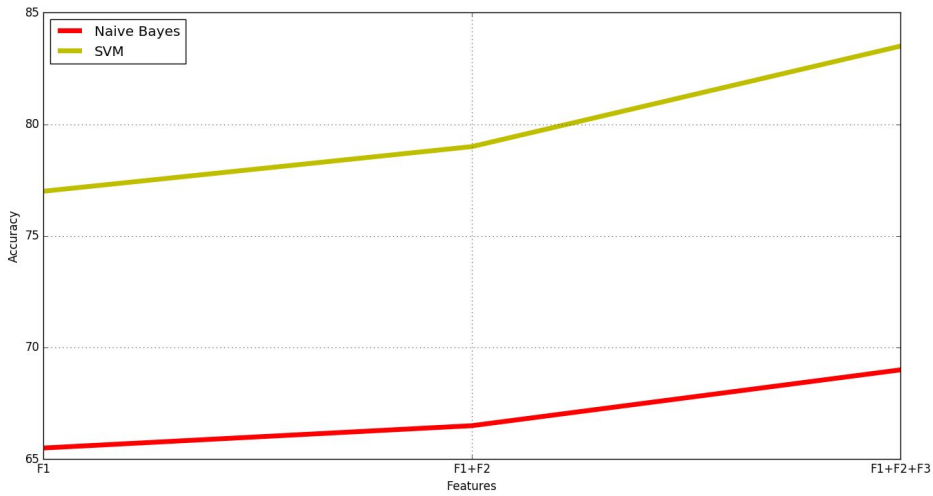

Figure 6: Comparison of features on the accuracy for Enron Dataset.

When lexical features alone were used, an accuracy of 65.5% and 77% was obtained for the Naive Bayes and SVM models. When combined with Syntactic features, the accuracy increased by only a negligible amount. One possible reason is that the messages in our datasets were too short to represent people's usage habits of function words. Another possible reason is that compared with the small number of words used in one message, the number of function words we used as features may be too large. When this set (Lexical + Syntactic), was combined with structural features, there was a considerable increase in the accuracy from 66.5% to 79% and from 79% to 83.5% for Naive Bayes and SVM respectively. It appears that an author's consistent writing patterns were reflected in the structural features. For example, some authors always write long sentences and paragraphs and some always use a greeting in every message while some use short sentences and write everything in a single paragraph while some others always use indentation at the beginning of the paragraph. Structural features appeared to be a good discriminator among authors for email messages.

But for the Reddit dataset, as mentioned in the previous section, the structural and syntactic features don't seem to be relevant. This is confirmed in the decrease in accuracy when these features are added as shown by the graph below:

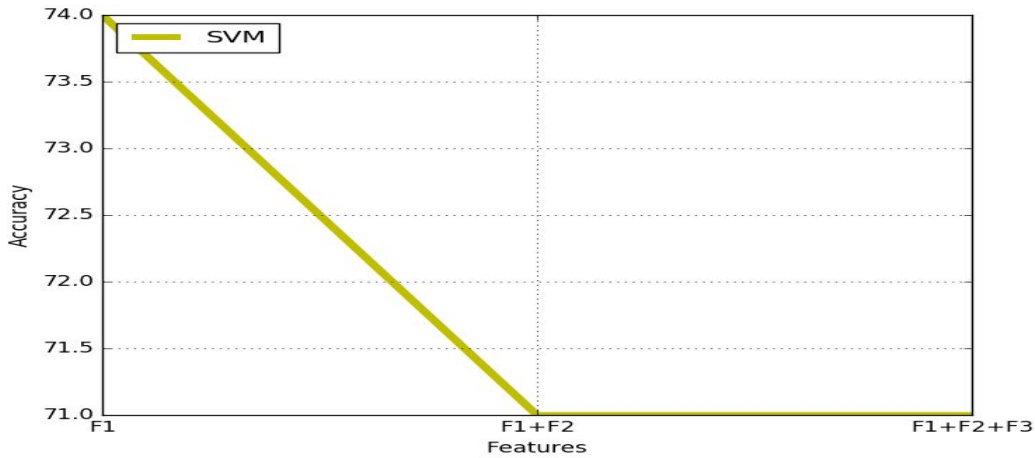

Figure 7: Decrease in accuracy when the feature sets are increased for Reddit Dataset.

c) Comparison of training size:

As the size of the training dataset was increased, the accuracy for SVM classifier also increased from 75% when 25 emails were used to 83.5% when 100 emails were used. This result was also consistent for the Reddit dataset which saw an increase in SVM accuracy

from 50% for 20 comments to 74% for 50 comments. Thus the accuracy of the classifier increases with increase in training data.

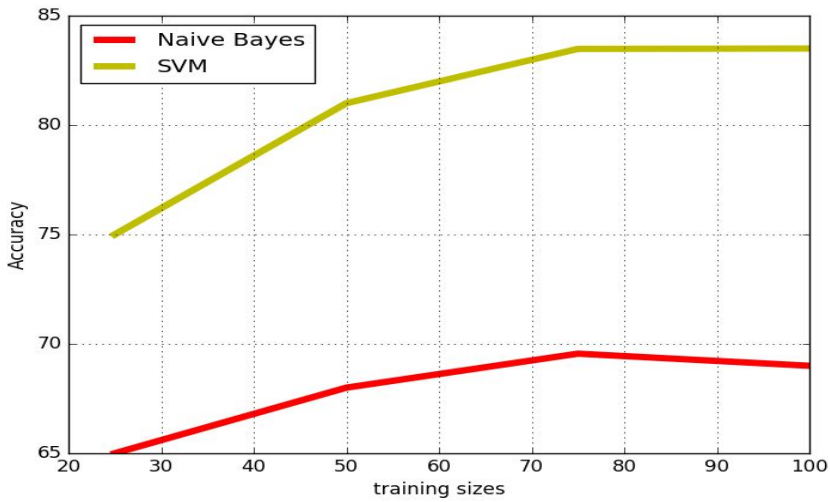

Figure 8: Comparison of number of messages per author with accuracy for Enron Dataset

d) Comparison of the number of authors:

As the number of authors increased, the accuracy of the SVM classifier decreased from 83.5% for 2 authors to 54.7% for 10 authors. This result was also consistent for the Reddit dataset which saw an decrease in SVM accuracy from 74% for 2 commenters to 42.5% for 5 commenters. Thus the accuracy of the classifier decreases with increase in number of authors(or classes).

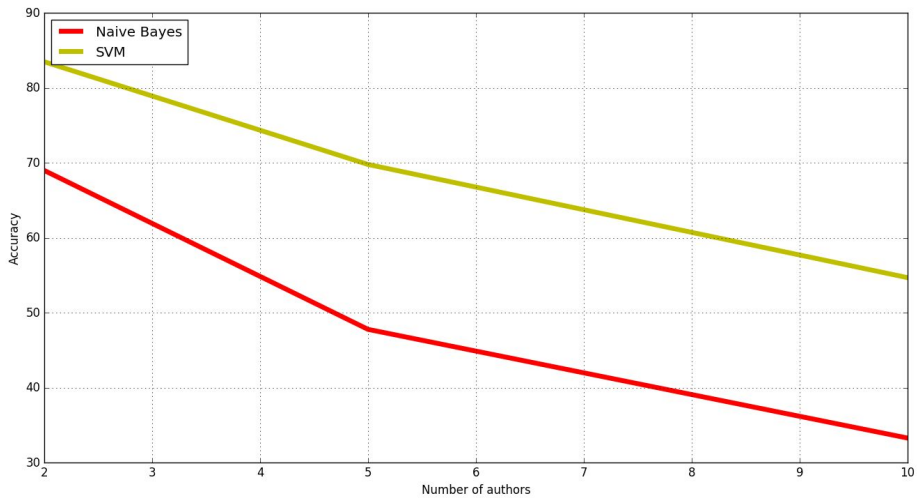

Figure 9: Decrease in accuracy for increase in number of authors for Enron Dataset.

DISCUSSION AND FUTURE WORK

In this project, we tried to replicate the work across the reference cited below – inspired by (Zheng et al., 2006)[1] for comparing different writing-style features to identify authorship in online messaging by implementing a framework for authorship identification of online messages. The effectiveness of this framework was evaluated by running experiments on Enron Email Dataset and Reddit Dataset. The results showed that the framework was capable in identifying the authors of the online messages/comments. We further changed the classification parameters and observed that structural and lexical features showed particular discriminating capabilities for authorship identification on emails. In case of very small text messages or comments, as present in the Reddit dataset, Lexical features alone are sufficient to discriminate between authors. SVM outperformed Naive Bayes significantly for the authorship-identification task. Different parameter settings of authorship identification had an impact on the performance and confirmed the speculations around the same.

One possible improvement to the project would be identifying the optimal set of features for online messaging. We could study how each feature contributes to the overall performance of the classifier and identify the minimal set of features which would provide a good accuracy. Secondly we could try and implement an unsupervised learning model to identify authors and their writings. This would reduce the effort of annotating the dataset. Thirdly, we could train the models developed in the project, on a bank of online comments made by known malicious authors. We could aim to design and deploy an automated intelligent scraper incorporated with the Support Vector Machine model to explore popular blogs and websites, to identify and collect other such online messages generated by the malicious authors in our database. By doing so we could probably help identifying all the aliases of each malicious author and create a comprehensive database of all such users online.

REFERENCES

[1] H. Chen, Z. Huang, J. Li, R. Zheng 2006. A framework for authorship Identification of Online Messages: writing-Style features and classification Techniques. JASIST, pp : 378- 393.

[2]J. Diederich, J. Kindermann, E. Leopold, G. Paass GMD Authorship Attribution with Support Vector Machines : Forschungszentrum Informationstechnik, D-52754 Sankt Augustin

[3]K. Tanaka-Ishii, S. Aihara 2015. Computational Constancy Measures of Texts—Yule's K and R´enyi's Entropy, Computational Linguistics, September 2015, Vol. 41, No. 3, Pages 481-502.

[4] de Vel 2000. Mining E-mail Authorship. Information Technology Division Defence Science and Technology Organisation, Australia.

[5]D.L. Hoover 2003, Another Perspective on Vocabulary Richness, Computers and the Humanities Vol. 37, No. 2 (May, 2003), pp. 151-178

[6]Stamatatos, Efstathios, Nikos Fakotakis, and Georgios Kokkinakis. "Computer-based authorship attribution without lexical measures." *Computers and the Humanities* 35.2 (2001): 193-214.

[7]Narayanan, Arvind, et al. ”On the feasibility of internet-scale author identification.” Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012.

[8]A. Castro and B. Lindauer. 2012. Author Identification on Twitter. Retrieved from http://cs229.stanford.edu/proj2012/CastroLindauerAuthorIdentificationOnTwitter.pdf

[9] S. Stanko, D. Lu, I. Hsu. Whose Book is it Anyway? Using Machine Learning to Identify the Author of Unknown Texts, Computer Science Department Stanford University.

[10] Houvardas, John, and Efstathios Stamatatos. "N-gram feature selection for authorship identification." *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer Berlin Heidelberg, 2006.