Author Identification

Beau Carlborg, Clara Richter, and Sunny Chiu

Update: Who we are, what we have been up to.

We are constructing an authorship identification which can be used to predict the most likely author of a given text.

Throughout this block, our project has undergone many changes.

Iteration 1: We spent the majority of time during this iteration retrieving our data. We found that finding reliable publication dates for a given date is really really hard.

Iteration 2: We realized that given the scope of this project, identifying the publication dates of works would be too large of an undertaking.

- We regathered our data and began the process of matching authors to their respective works
- We started collecting lexical stylistic features for our matrix.

Iteration 3 (goal): Expand our data feature set to include n-grams and parts of speech grams, and to implement and run a model on our classification.

Constructing a Complete Row

For this iteration, it was a significant goal of ours to incorporate a larger number of data features into our model, while keeping data sets at a reasonable size.

Desired Feature Set

Lexical Features,

Distribution of Word Length, Distribution of Sentence Length, Word Richness, etc.

N-Grams,

Relative Frequency of mono-grams (one word chunks), bi-grams (two word chunks), and tri-grams (three word chunks).

POS-Grams

Relative frequency of Parts of speech grams in one word, two word, and three word chunks.

Hey guys… That seems like it is gonna be a lot of data!

And that’s because it is!

Excerpt of N-grams and POS-grams

‘CC’	'VB',	'accidental',	'agricultural',
'CD',	'VBD',	'accidentally',	'aguecheek',
'DT',	'VBG',	'accidentals',	'agues',
'EX',	'VBN',	'accidently',	'ah',
'FW',	'VBP',	'accidents',	'ah-hah-lah-nih',
'IN',	'VBZ',	'accidia',	'ahab',
'JJ',	'WDT',	'accommodate',	'ahead',
'JJR',	'WP',	'accommodated',	'ahem',
'JJS',	'WP$',	'accommodating',	'ahold',
'LS',	'WRB',	'accommodations',	'ahoy',
'MD',	'[',	'accompanied',	'ai',
'NN',	'[8',	'accompanies',	"ain't",
'NNP',	']',	'accompaniment',	'ainsi',
'NNPS',	];',	'accompany',	'air',
'NNS',	'a',	'accompanying',	"air's",
'PDT',	"a'most",	'accomplice',	'aira',
'POS',	"a'thinkin,	'accomplices',	'aired',
'PRP',	'a-all',	'accomplish',	'airfields',
'PRP$',	'a-bloom'	'accomplished',	'airily',
'RB',	'a-comin'	'accomplishes',	'airing',
'RBR',	'a-coming’	'accomplishment',	'airline',
'RBS',	'a-croakin'	'accomplishments'	'airliner',
'RP',	'a-dying',	'accord',	'airplane',
'SYM',	'a-feared',	'accordance',	'Airport',

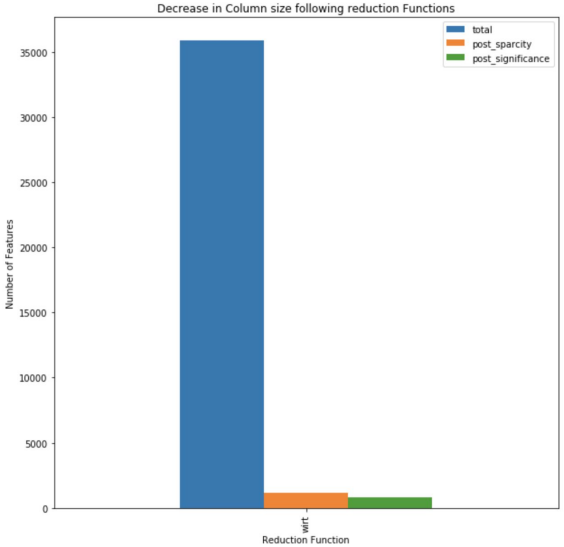
Challenge 1: How to reduce Dimensionality

- In order to reduce dimensionality, we created two functions which can be used to reduce the number of columns.
  - reduce_sparce_columns( )
  - reduce_similar_columns( )

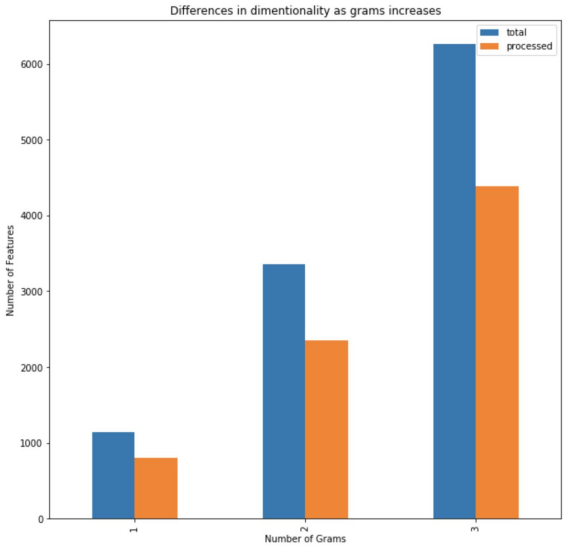
Books	Author	N-Gram: “jeepers”	N-Gram 1: “polywog”
The Cat in the Hat	Dr. Suess	0.043	0.0
Green Eggs and Ham	Dr. Suess	0.0024	0.0
One Fish Two Fish	Dr. Suess	0.0232	0.0
The Sneetches	Dr. Suess	0.0	0.0
Fox in Socks	Dr. Suess	0.062	0.0
Harry Potter 1	J.K. Rowling	0.03	0.0
Harry Potter 2	J.K. Rowling	0.0001	0.0
Harry Potter 3	J.K. Rowling	0.067	0.0
Harry Potter 4	J.K. Rowling	0.00342	0.0000050
Harry Potter 5	J.K. Rowling	0.00232	0.0

Reducing the number of features

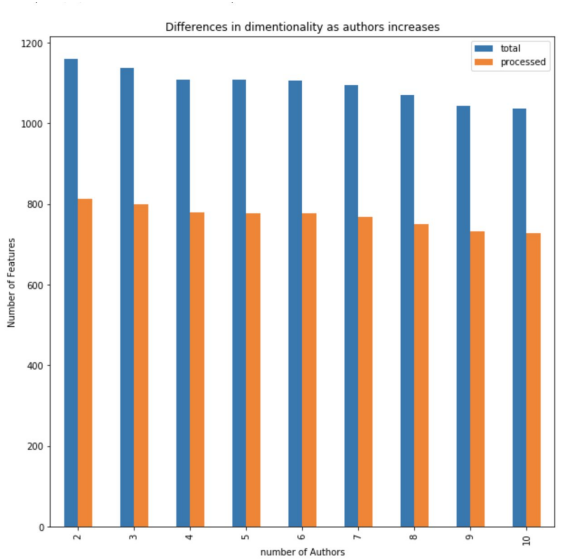
Reduction in Total Features



Features as N-Grams Increases



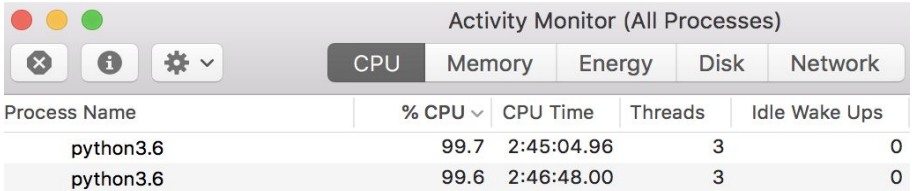
Features as Authors Increase



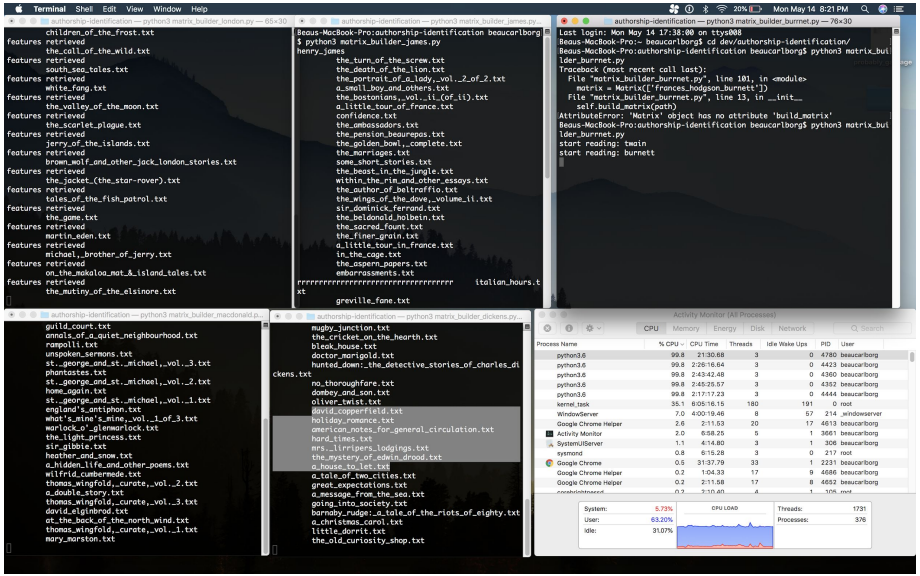
Challenge 2: Big Data is Slow Data

Even though we came up with an effective method for removing superfluous data, we still struggled with the runtime for most of our data processing.

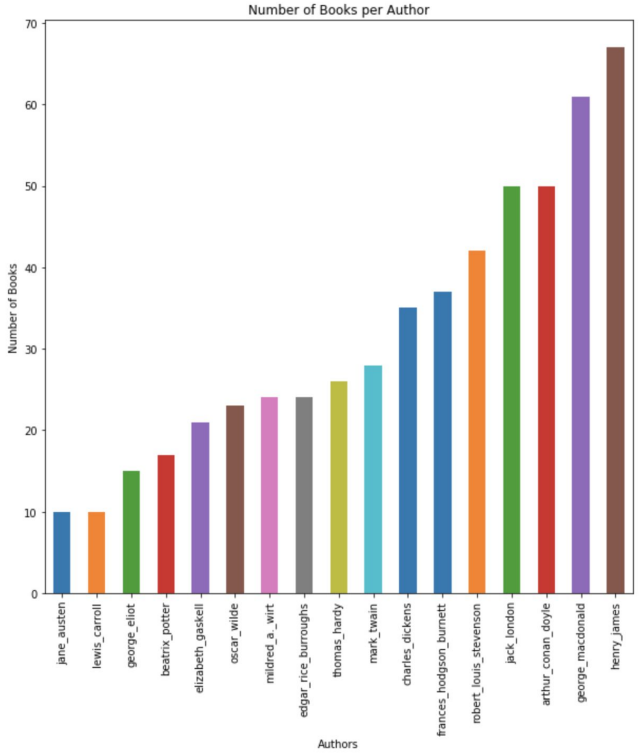
This was a lot of fun, everyone's computer fans were on full, and CPU temperatures floated around 160 degrees.

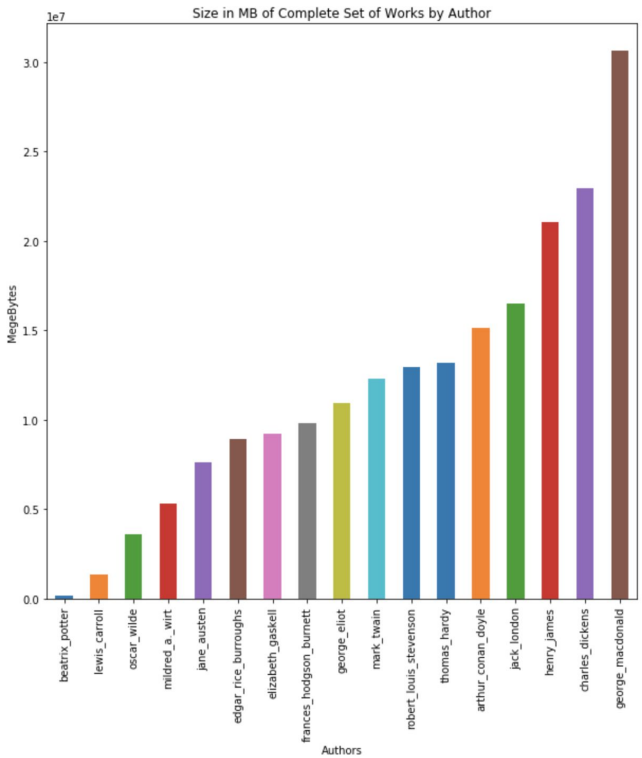


Process Name	% CPU	CPU Time	Threads	Idle Wake Ups
python3.6	99.7	2:45:04.96	3	0
python3.6	99.6	2:46:48.00	3	0
python3.6	99.6	2:18:39.69	3	0
python3.6	99.5	22:53.05	3	0
python3.6	99.5	2:27:39.15	3	0

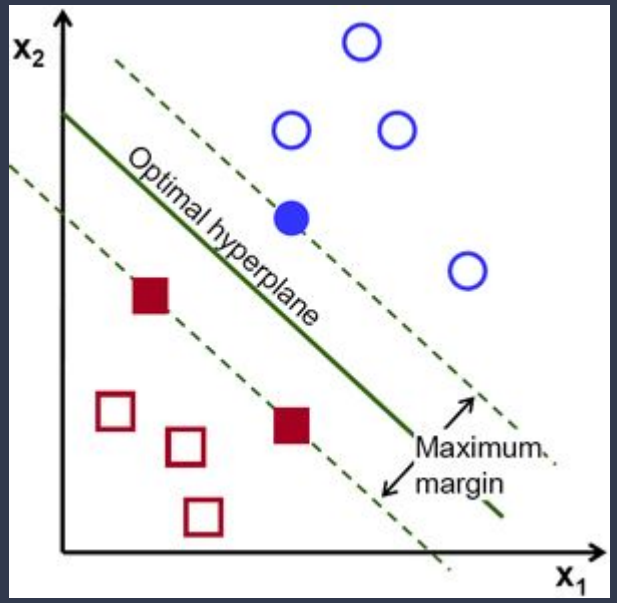


Choosing Authors to Test Based on Data Size





Our Model Choice



The conceptual idea behind SVM classification

The nature of our problem:
discrete supervised classification

What other researchers have used:

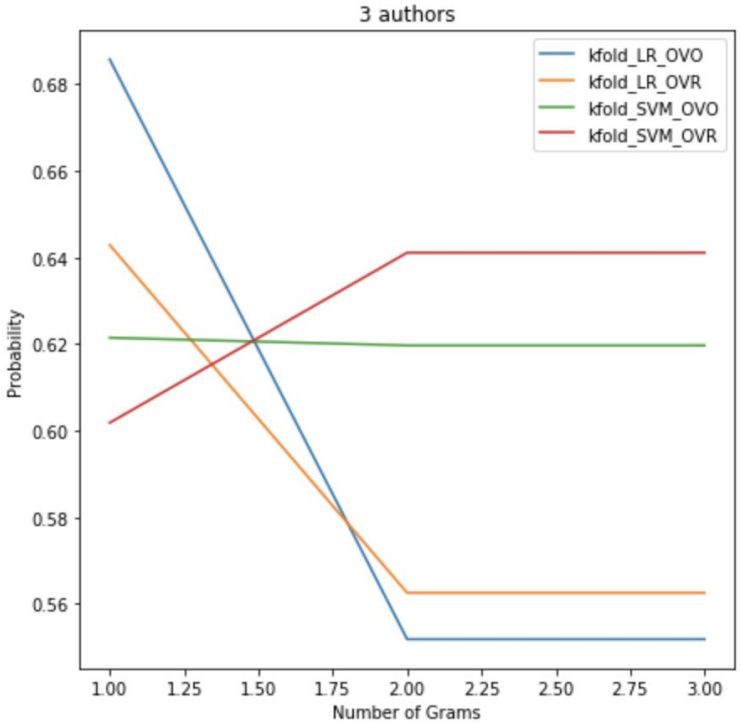
1) Support Vector Machines
2) Artificial Neural Networks
3) k-Nearest Neighbour
4) Decision Trees
5) Random Forests

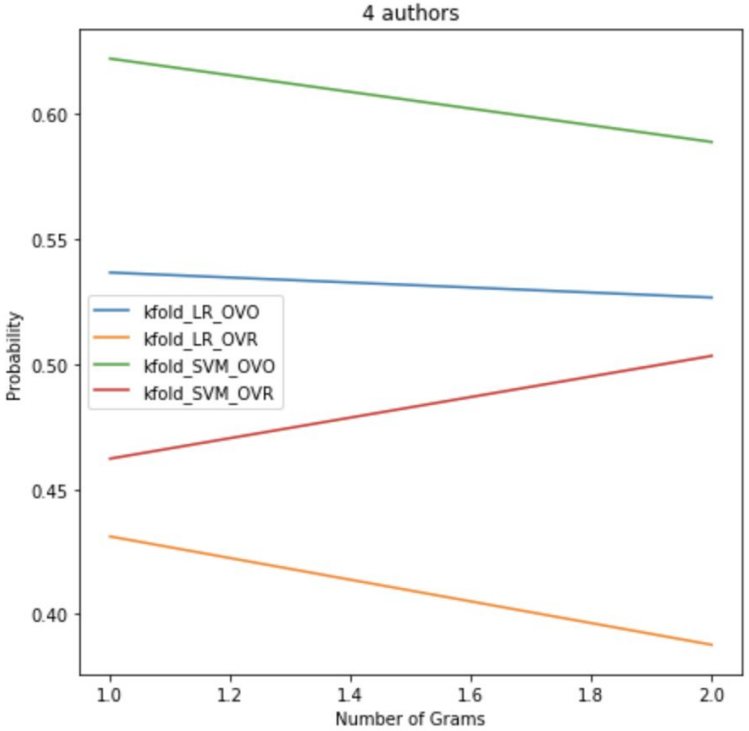
What we decided to use:

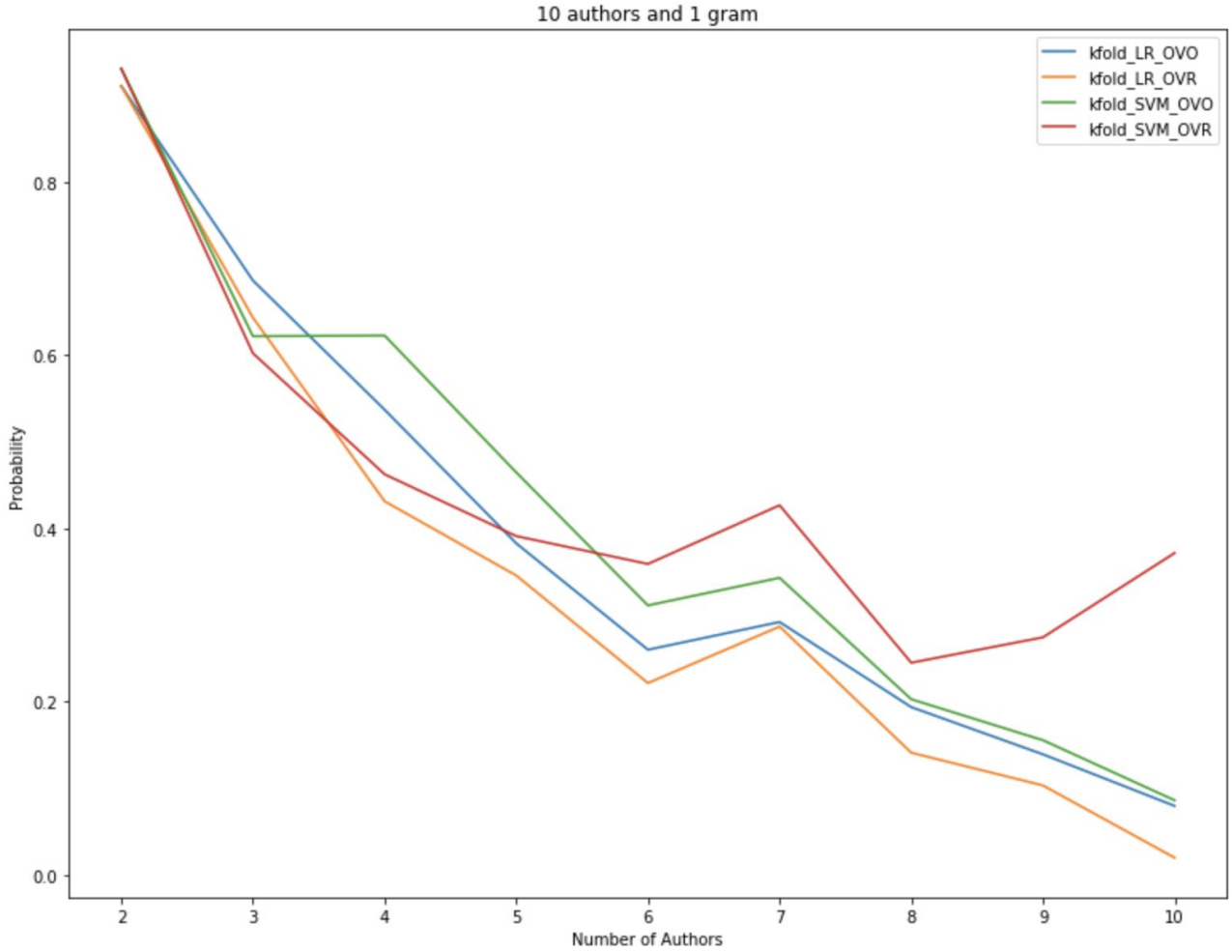
1) Logistic Regression (one vs. one, one vs. rest)
    a) The classification model we each had the most thorough understanding of.
2) Support Vector Machines (one vs. one, one vs. rest)
    a) The de-facto model for image recognition and speech recognition.
    b) Know for robustness with high dimensional data

We used 10-fold cross validation for handling training and testing

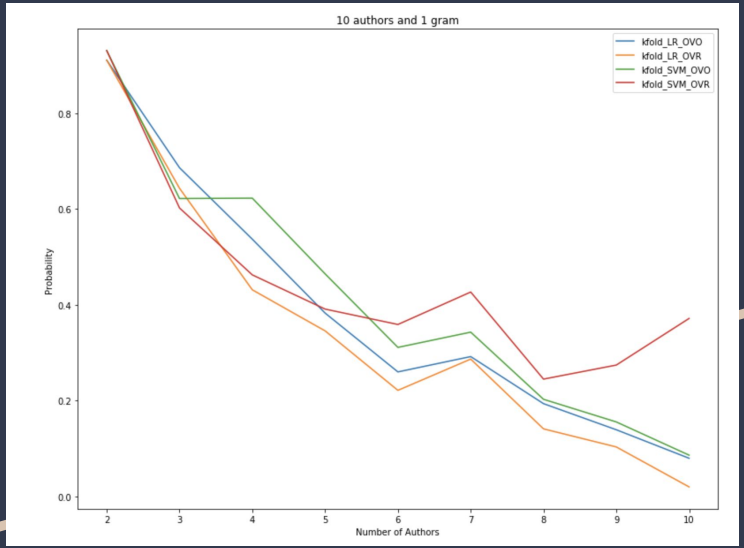
Model Results






10 authors and 1 gram
kfold_LR_OVO
kfold_LR_OVR
kfold_SVM_OVO
kfold_SVM_OVR
Probability
0.0
0.2
0.4
0.6
0.8
2
3
4
5
6
7
8
9
10
Number of Authors

Interpreting our Results



Observations

- One grams typically had higher prediction accuracy.
    - We assume this higher prediction capability is due to overfitting when using more grams.

The Best Models

- 5 > authors > 2 : One Vs One Support Vector Machine
- Authors > 5: One Vs Rest Support Vector Machine

Future Work

Due to short iteration cycles of this project, we spent an extensive amount of our time reworking previous iterations, cleaning the data and building our domain knowledge.

For future work, we assume we would have access to a more powerful cluster of computers allowing more extensive data collection and processing.

1) Try out more models (k-Nearest Neighbour specifically)
2) Delve into the many variations of our models we could implement in order to fine tune **for our specific problem.**
    a) For Example, changing the kernel functions, using more fine tuned training functions.
3) With more domain knowledge, we would use more representative, and better refined, features
    a) “Function words”
    b) Longer parts-of-speech ngrams!
    c) Metrics that capture patterns in sentence structure better
4) Authorship identification on other types of texts (rather than classic English books)