Lab 5 Report

Information Retrieval and Text Mining

Authors:

Matthew Jaojoco (mjaojoco@calpoly.edu)

Joshua Rauvola (jrauvola@calpoly.edu)

Abstract

The following report outlines our findings when attributing authorship to authors using two classifying algorithms, K Nearest Neighbors and Random Forests. In order to attribute authorship we vectorized representations of news stories from the well-known Reuters 50-50 dataset. For each of the classifiers, we attempt to classify these stories with the highest possible accuracy by tuning the parameters for the respective algorithm and analyzing our results.

Introduction

Given only the news stories to start with, we implemented a program that turns each story into a vectorized representation of words in the particular story, after preprocessing by filtering out a provided list of stopwords and stemming. We then take these vector representations and give them to one of our classifiers: KNN or Random Forest which implements C45. For both of our classifiers, we provide the resulting calculations: Hits being true positives for correct attribution, Misses being false positives for correct attribution, Strikes being false positives for correct attribution, Precision which gives us the percentage of documents retrieved that were relevant, Recall which gives us out of all relevant documents possible what percentage was actually retrieved and F-Measure. F-Measure is the variable we used to make our final conclusion on which author was the easiest to evaluate overall as it takes into account both precision and recall. We also evaluate the overall accuracy of the whole algorithm to evaluate between matrices and algorithms.

Implementation

Preprocessing/Vectorizing

The goal of this experiment was to understand the Reuters 50-50 dataset by analyzing the dataset with two different algorithms, K Nearest Neighbors and Random Forests and to see if those algorithms would be able to predict which document belonged to which author. To be able to generate the needed forest and clustering of data points we first needed to vectorize and produce different distance matrices to apply to the algorithms.

Firstly, for both of the datasets after iterating and reading in each file we were able to generate a dataset of all the files. These datasets were constrained by the metrics of stopword files and stemming (True vs. False) which either cut down on the words generated by the Reuters 50-50 dataset or left them be. Stop words are typically words that are extremely common and that would appear to be of little value in helping the algorithms in being able to predict which author belonged to which article.

Next we generated the frequency of each of the words that occurred in a specific document and how many times that word showed up in all documents overall. After storing these two variables we were finally able to generate the values needed for RF and KNN. The first metric that we needed was tf-idf or Term Frequency – Inverse Document Frequency. This metric reflects how important a word is to a document in a collection like ours. We used it as a weighing factor for when we generated our trees for the RF algorithm.

Lastly, we generate two distance matrices for KNN, one using the cosine similarity equation and the other using the okapi equation. The cosine similarity equation isn't an overly complicated equation but it is used to find the similarity between two vectors of an inner product space. Since KNN relies on distance or similarity between two vectors it does fit nicely with this experiment. The next equation we used was a much more complex one but is more commonly used as a ranking function which gets used by search engines to estimate the relevance of documents to a given search query. We used these two distance metrics later for KNN and contrasted how they worked overall with different preprocessed metrics. Our implementation can be found in preprocessing.py

K Nearest Neighbors

Our implementation of K Nearest Neighbors can be found in knn_implementation.py. This algorithm is a non-parametric classification method and is typically used for classification and regression. As we are dealing with a classification problem in trying to find out which document belonged to which author the algorithm fit well with that goal. Since we had generated the distance matrices previously in the preprocessing method the implementation of this algorithm mainly involved two actions comparing each vector to the matrices and finding the optimal k value.

The k value in the instance of KNN is used in the algorithm to generate the amount of neighbors that are closest to the vector or document that is being evaluated. After the amount of neighbors has been generated, finding the most common value in the amount generated is needed to assign that document you are evaluating the most common value. Once the most common value is assigned to that vector or document, the KNN algorithm continues until all of the documents have been predicted for. Once the predictions have been generated they can be compared to the actual authors of the documents to be able to give an accurate understanding of how well the dataset ran on that specific matrix. We then hypertuned our parameters to evaluate the best set of parameters given the distance metric between okapi and cosine similarities, the amount of stopwords, and if stemming was turned on or not.

Random Forest

Our implementation of Random Forest can be found in RFAuthorship.py. The algorithm gets its implementation of C45 from InduceC45.py. In this algorithm, the set of stories is repeatedly partitioned so long as the uniformity (entropy) of the data improves greater than a predetermined threshold. Random Forest creates random subsets of the data, and creates N decision trees using the C45 algorithm. C45 partitions the data based on a selected attribute. With the Reuters 50-50 dataset, this selected attribute will always be the tf-idf of a certain word in the total Vocabulary of the set. By grouping together stories that have similar weights of the same words, C45 hopes

to increase its ability to predict each story's class. Once the trees are built, they are traversed and the most common result of the traversal becomes the prediction. RFAuthorship then evaluates its own results by constructing and outputting an overall confusion matrix, as well as overall accuracy results and individual accuracy/precision metrics with respect to each author in the dataset.

Tuning Process and Result Analysis

Preprocessing / Vectorizing

When preprocessing the news stories and creating their vectors, there were two main choices that we were faced with. Whether or not to filter out stopwords, and whether or not to include stemming. To evaluate these two metrics we ran all possible combinations in the generation of through all 5000 documenta that allowed us to exhaust all possible ways that these metrics could be constructed. We also implemented this process for both distance calculations of okapi and cosine and for the tf-idf equation as well. By having all of these matrices created and stored we were more effectively able to hypertune our parameters for both algorithms KNN and RF.

K Nearest Neighbors

For KNN, we tuned the algorithm's only parameter k by testing multiple k values for each possible list of stopwords. We found that k ranging anywhere from 3 to 23 has minimal impact on the accuracy of KNN's predictions, but k = 1 provides an accuracy significantly higher than 3<k<23. We did not expect this to happen, but we believe in the correctness of the results our program is outputting. Perhaps each file in the testing set is closely similar to a file in the training set by the same author, and that is often the one neighbor being matched. Or maybe this is just a characteristic of the dataset we were provided.

We also tested two different similarity values: cosine similarity and okapi. After looking at the results from the different combinations we tried, it appears that our KNN implementation is more accurate when using cosine similarity as the similarity metric. Independent of both k and similarity metric, it appears that there are authors notably easy to recognize, namely Kevin Drawbaugh and Peter Humphrey. There are also specific authors that are notably difficult to recognize, namely Pierre Tran and Sarah Davidson.

Our KNN implementation also seems to have strange behavior when handling an empty list of stopwords. In these runs, our implementation credits the vast majority of stories to Eric Auchard with most other authors having a miss rate of 100% and sometimes even predicting Eric as the author for all 5000 documents. One hypothesis for this phenomenon is that Eric has a limited vocabulary and a significant amount of the words he uses are common stopwords. This would increase his similarity to each document and thus make it harder to detect the unique words that would typically allow our algorithm to attribute a news story to its proper author.

Random Forest

For Random Forest, the parameters we chose to tune were the attributes to be included in the creation of each decision tree (k) as well as the number of overall trees that we create in the forest(N). The values we tested for k were 10, 15, and 20 and decided that each attribute would represent the tf-idf of a word in the document since it has some notion of weight per word, as opposed to raw frequencies. We respectively paired each of these values for k to 4000, 2667, and 2000 for N. These values ensure that we maintain the desired property of $(N * k) > |A|$. We decided to keep the threshold as a control set at 0.1 and also kept the number of datapoints to be included in the decision trees (m) as 500 for every run. We then ran these parameters on each of the possible stopword lists.

The vast majority of the implementation of random forests was adopted from lab 3, with removal of unneeded steps, such as checking whether an attribute is categorical or numerical since all attributes in A are numerical with the exception of the class label, or author. Additionally, multiple iterative loops in the code were switched to list comprehensions along with other minor changes that attempt to minimize runtime considering that our list of attributes is around 30,000 to 40,000 depending on which stopword list is selected and whether stemming is executed.

We found that the accuracies for Random Forest are significantly higher when stemming is included in the preprocessing and vectorization step, with a few exceptions to the trend. Additionally, we discovered that in our runs, N and k individually have little impact on accuracy but keeping the property of $(N * k) > |A|$ is useful in conducting accurate predictions.

Comparative Analysis

Typically, we can expect Random Forest to produce more accurate and precise results provided the parameters are finely tuned to the specific dataset being worked on. The algorithm theoretically does a better job of providing a holistic view of the data. Considering that each author has 100 papers in the set, it would make sense that finding the few closest neighbors

wouldn't always provide an accurate picture. Say for example that two authors interviewed the same person who gave similar quotes to both authors, then the subject matter would have more influence on the story than the author, whom we are trying to identify.

KNN has advantages in its simplicity and ease of implementation. Our implementation of KNN also has a significant advantage over our Random Forest implementation in terms of runtime, but on average provides slightly less accurate predictions. In our findings it seems as though KNN requires less tuning as there are less parameters to hypertune and changing the k parameter seems to have minimal impact with small k values. Due to this, it is more difficult to gain accuracy by tuning parameters but it is also harder to get noticeably inaccurate results from.

However, provided that you have the time and capacity to tune the parameters of Random Forest to the dataset and test different results, then Random Forest is the classifying method that we would recommend. Its highest accuracy is better than KNN's highest accuracy, and Random Forest was able to handle stopword/stemming conditions where KNN failed (no stopwords and no stemming). Random Forest also provided a wider range of authors that it could easily identify, using f-measure as the metric for ease of identification. An unlucky selection of random points and attributes could still cause poor results even with parameter tuning, but Random Forest's ability to be adjusted to the model and output of higher average accuracy makes up for its downfalls.

Appendix

K Nearest Neighbors

Vectorizer Values: Stopwords = empty, Stemming = False
KNN Values: K = 15, similarity measure = okapi

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
5	EricAuchard	100	0	4846	0.020218	1.0	0.039635
0	RobinSidel	0	100	0	0.000000	0.0	0.000000
38	MarkBendeich	0	100	0	0.000000	0.0	0.000000
28	MureDickie	0	100	0	0.000000	0.0	0.000000
29	RogerFillion	0	100	23	0.000000	0.0	0.000000
30	JimGilchrist	0	100	0	0.000000	0.0	0.000000
31	BradDorfman	0	100	0	0.000000	0.0	0.000000
32	AlanCrosby	0	100	31	0.000000	0.0	0.000000
33	JonathanBirt	0	100	0	0.000000	0.0	0.000000
34	BenjaminKangLim	0	100	0	0.000000	0.0	0.000000
35	ThieresePoletti	0	100	0	0.000000	0.0	0.000000
36	KeithWeir	0	100	0	0.000000	0.0	0.000000
37	JoWinterbottom	0	100	0	0.000000	0.0	0.000000
39	JaneMacartney	0	100	0	0.000000	0.0	0.000000
26	PierreTran	0	100	0	0.000000	0.0	0.000000
40	MatthewBunce	0	100	0	0.000000	0.0	0.000000
41	ToddNissen	0	100	0	0.000000	0.0	0.000000
42	PeterHumphrey	0	100	0	0.000000	0.0	0.000000
43	TimFarrand	0	100	0	0.000000	0.0	0.000000
44	SarahDavison	0	100	0	0.000000	0.0	0.000000
45	GrahamEarnshaw	0	100	0	0.000000	0.0	0.000000
46	BernardHickey	0	100	0	0.000000	0.0	0.000000
47	KirstinRidley	0	100	0	0.000000	0.0	0.000000
48	AlexanderSmith	0	100	0	0.000000	0.0	0.000000
27	HeatherScoffield	0	100	0	0.000000	0.0	0.000000
25	TanEeLyn	0	100	0	0.000000	0.0	0.000000
1	LynnleyBrowning	0	100	0	0.000000	0.0	0.000000
24	WilliamKazer	0	100	0	0.000000	0.0	0.000000
2	KouroshKarimkhany	0	100	0	0.000000	0.0	0.000000
3	MichaelConnor	0	100	0	0.000000	0.0	0.000000
4	JoeOrtiz	0	100	0	0.000000	0.0	0.000000
6	AaronPressman	0	100	0	0.000000	0.0	0.000000
7	SimonCowell	0	100	0	0.000000	0.0	0.000000
8	LynneO'Donnell	0	100	0	0.000000	0.0	0.000000
9	EdnaFernandes	0	100	0	0.000000	0.0	0.000000

10	KevinMorrison	0	100	0	0.000000	0.0	0.000000
11	SamuelPerry	0	100	0	0.000000	0.0	0.000000
12	PatriciaCommins	0	100	0	0.000000	0.0	0.000000
13	JohnMastrini	0	100	0	0.000000	0.0	0.000000
14	JanLopatka	0	100	0	0.000000	0.0	0.000000
15	KevinDrawbaugh	0	100	0	0.000000	0.0	0.000000
16	KarlPenhaul	0	100	0	0.000000	0.0	0.000000
17	MartinWolk	0	100	0	0.000000	0.0	0.000000
18	ScottHillis	0	100	0	0.000000	0.0	0.000000
19	DavidLawder	0	100	0	0.000000	0.0	0.000000
20	FumikoFujisaki	0	100	0	0.000000	0.0	0.000000
21	MarcelMichelson	0	100	0	0.000000	0.0	0.000000
22	NickLouth	0	100	0	0.000000	0.0	0.000000
23	DarrenSchuettler	0	100	0	0.000000	0.0	0.000000
49	LydiaZajc	0	100	0	0.000000	0.0	0.000000

Matrix Totals

Best Author: EricAuchard
Correct: 100
Incorrect: 4900
Accuracy: 0.02

Vectorizer Values: Stopwords = long, Stemming = False
KNN Values: K = 1, similarity measure = cosine

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
15	KevinDrawbaugh	98	2	6	0.942308	0.98	0.960784
46	BernardHickey	97	3	6	0.941748	0.97	0.955665
42	PeterHumphrey	94	6	5	0.949495	0.94	0.944724
13	JohnMastrini	94	6	9	0.912621	0.94	0.926108
4	JoeOrtiz	93	7	8	0.920792	0.93	0.925373
0	RobinSidel	91	9	6	0.938144	0.91	0.923858
5	EricAuchard	97	3	15	0.866071	0.97	0.915094
29	RogerFillion	90	10	16	0.849057	0.90	0.873786
1	LynnleyBrowning	88	12	17	0.838095	0.88	0.858537
16	KarlPenhaul	82	18	10	0.891304	0.82	0.854167
27	HeatherScoffield	81	19	9	0.900000	0.81	0.852632
33	JonathanBirt	80	20	8	0.909091	0.80	0.851064
48	AlexanderSmith	82	18	11	0.881720	0.82	0.849741
32	AlanCrosby	89	11	21	0.809091	0.89	0.847619
7	SimonCowell	88	12	20	0.814815	0.88	0.846154
38	MarkBendeich	85	15	17	0.833333	0.85	0.841584
35	ThieresePoletti	82	18	15	0.845361	0.82	0.832487

11	SamuelPerry	84	16	18	0.823529	0.84	0.831683
8	LynneO'Donnell	83	17	17	0.830000	0.83	0.830000
45	GrahamEarnshaw	86	14	22	0.796296	0.86	0.826923
41	ToddNissen	76	24	9	0.894118	0.76	0.821622
2	KouroshKarimkhany	91	9	31	0.745902	0.91	0.819820
18	ScottHillis	81	19	17	0.826531	0.81	0.818182
40	MatthewBunce	84	16	23	0.785047	0.84	0.811594
49	LydiaZajc	71	29	8	0.898734	0.71	0.793296
39	JaneMacartney	77	23	20	0.793814	0.77	0.781726
30	JimGilchrist	80	20	25	0.761905	0.80	0.780488
31	BradDorfman	73	27	16	0.820225	0.73	0.772487
43	TimFarrand	73	27	16	0.820225	0.73	0.772487
20	FumikoFujisaki	78	22	24	0.764706	0.78	0.772277
22	NickLouth	76	24	22	0.775510	0.76	0.767677
6	AaronPressman	79	21	28	0.738318	0.79	0.763285
12	PatriciaCommins	75	25	23	0.765306	0.75	0.757576
24	WilliamKazer	83	17	37	0.691667	0.83	0.754545
34	BenjaminKangLim	79	21	31	0.718182	0.79	0.752381
21	MarcelMichelson	74	26	23	0.762887	0.74	0.751269
28	MureDickie	84	16	40	0.677419	0.84	0.750000
10	KevinMorrison	82	18	37	0.689076	0.82	0.748858
47	KirstinRidley	67	33	12	0.848101	0.67	0.748603
36	KeithWeir	77	23	42	0.647059	0.77	0.703196
23	DarrenSchuettler	64	36	21	0.752941	0.64	0.691892
37	JoWinterbottom	64	36	23	0.735632	0.64	0.684492
19	DavidLawder	57	43	19	0.750000	0.57	0.647727
25	TanEeLyn	71	29	53	0.572581	0.71	0.633929
17	MartinWolk	53	47	28	0.654321	0.53	0.585635
14	JanLopatka	52	48	36	0.590909	0.52	0.553191
9	EdnaFernandes	55	45	61	0.474138	0.55	0.509259
26	PierreTran	38	62	34	0.527778	0.38	0.441860
44	SarahDavison	50	50	80	0.384615	0.50	0.434783
3	MichaelConnor	37	63	40	0.480519	0.37	0.418079

Matrix Totals

Best Author: KevinDrawbaugh
Correct: 3865
Incorrect: 1135
Accuracy: 0.773

Vectorizer Values: Stopwords = long, Stemming = False
KNN Values: K = 9, similarity measure = cosine

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
42	PeterHumphrey	97	3	6	0.941748	0.97	0.955665
15	KevinDrawbaugh	97	3	7	0.932692	0.97	0.950980
5	EricAuchard	99	1	24	0.804878	0.99	0.887892
46	BernardHickey	98	2	28	0.777778	0.98	0.867257
4	JoeOrtiz	89	11	20	0.816514	0.89	0.851675
13	JohnMastrini	93	7	31	0.750000	0.93	0.830357
27	HeatherScoffield	74	26	7	0.913580	0.74	0.817680
0	RobinSidel	77	23	19	0.802083	0.77	0.785714
32	AlanCrosby	84	16	30	0.736842	0.84	0.785047
49	LydiaZajc	68	32	6	0.918919	0.68	0.781609
2	KouroshKarimkhany	92	8	45	0.671533	0.92	0.776371
1	LynnleyBrowning	83	17	34	0.709402	0.83	0.764977
11	SamuelPerry	84	16	37	0.694215	0.84	0.760181
48	AlexanderSmith	76	24	24	0.760000	0.76	0.760000
34	BenjaminKangLim	82	18	34	0.706897	0.82	0.759259
29	RogerFillion	79	21	32	0.711712	0.79	0.748815
33	JonathanBirt	74	26	26	0.740000	0.74	0.740000
20	FumikoFujisaki	83	17	44	0.653543	0.83	0.731278
12	PatriciaCommins	71	29	25	0.739583	0.71	0.724490
28	MureDickie	82	18	46	0.640625	0.82	0.719298
38	MarkBendeich	70	30	27	0.721649	0.70	0.710660
18	ScottHillis	65	35	20	0.764706	0.65	0.702703
8	LynneO'Donnell	65	35	21	0.755814	0.65	0.698925
45	GrahamEarnshaw	71	29	34	0.676190	0.71	0.692683
24	WilliamKazer	81	19	54	0.600000	0.81	0.689362
41	ToddNissen	64	36	22	0.744186	0.64	0.688172
16	KarlPenhaul	62	38	22	0.738095	0.62	0.673913
47	KirstinRidley	58	42	16	0.783784	0.58	0.666667
35	ThieresePoletti	66	34	32	0.673469	0.66	0.666667
7	SimonCowell	71	29	43	0.622807	0.71	0.663551
6	AaronPressman	74	26	52	0.587302	0.74	0.654867
22	NickLouth	55	45	14	0.797101	0.55	0.650888
43	TimFarrand	57	43	22	0.721519	0.57	0.636872
10	KevinMorrison	76	24	65	0.539007	0.76	0.630705
21	MarcelMichelson	63	37	39	0.617647	0.63	0.623762
40	MatthewBunce	68	32	55	0.552846	0.68	0.609865
30	JimGilchrist	52	48	20	0.722222	0.52	0.604651
23	DarrenSchuettler	55	45	30	0.647059	0.55	0.594595
36	KeithWeir	57	43	38	0.600000	0.57	0.584615
39	JaneMacartney	50	50	31	0.617284	0.50	0.552486
37	JoWinterbottom	44	56	19	0.698413	0.44	0.539877
19	DavidLawder	43	57	18	0.704918	0.43	0.534161
31	BradDorfman	44	56	27	0.619718	0.44	0.514620
25	TanEeLyn	63	37	103	0.379518	0.63	0.473684
9	EdnaFernandes	53	47	89	0.373239	0.53	0.438017
17	MartinWolk	33	67	26	0.559322	0.33	0.415094

14	JanLopatka	36	64	48	0.428571	0.36	0.391304
3	MichaelConnor	33	67	38	0.464789	0.33	0.385965
44	SarahDavison	32	68	70	0.313725	0.32	0.316832
26	PierreTran	17	83	20	0.459459	0.17	0.248175

Matrix Totals

Best Author: PeterHumphrey
Correct: 3360
Incorrect: 1640
Accuracy: 0.672

Vectorizer Values: Stopwords = long, Stemming = False
KNN Values: K = 15, similarity measure = cosine

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
15	KevinDrawbaugh	95	5	10	0.904762	0.95	0.926829
42	PeterHumphrey	90	10	8	0.918367	0.90	0.909091
5	EricAuchard	99	1	29	0.773438	0.99	0.868421
4	JoeOrtiz	86	14	14	0.860000	0.86	0.860000
46	BernardHickey	97	3	31	0.757812	0.97	0.850877
32	AlanCrosby	85	15	18	0.825243	0.85	0.837438
49	LydiaZajc	72	28	7	0.911392	0.72	0.804469
1	LynnleyBrowning	86	14	33	0.722689	0.86	0.785388
27	HeatherScoffield	70	30	11	0.864198	0.70	0.773481
11	SamuelPerry	82	18	31	0.725664	0.82	0.769953
13	JohnMastrini	93	7	49	0.654930	0.93	0.768595
34	BenjaminKangLim	84	16	35	0.705882	0.84	0.767123
2	KouroshKarimkhany	87	13	40	0.685039	0.87	0.766520
48	AlexanderSmith	77	23	25	0.754902	0.77	0.762376
20	FumikoFujisaki	85	15	42	0.669291	0.85	0.748899
33	JonathanBirt	77	23	29	0.726415	0.77	0.747573
0	RobinSidel	73	27	23	0.760417	0.73	0.744898
38	MarkBendeich	70	30	19	0.786517	0.70	0.740741
8	LynneO'Donnell	70	30	19	0.786517	0.70	0.740741
45	GrahamEarnshaw	77	23	35	0.687500	0.77	0.726415
29	RogerFillion	79	21	43	0.647541	0.79	0.711712
18	ScottHillis	62	38	20	0.756098	0.62	0.681319
43	TimFarrand	64	36	24	0.727273	0.64	0.680851
28	MureDickie	82	18	60	0.577465	0.82	0.677686
41	ToddNissen	63	37	23	0.732558	0.63	0.677419
12	PatriciaCommins	64	36	26	0.711111	0.64	0.673684
21	MarcelMichelson	63	37	25	0.715909	0.63	0.670213
6	AaronPressman	71	29	42	0.628319	0.71	0.666667
24	WilliamKazer	78	22	57	0.577778	0.78	0.663830

7	SimonCowell	71	29	44	0.617391	0.71	0.660465
22	NickLouth	60	40	24	0.714286	0.60	0.652174
47	KirstinRidley	56	44	20	0.736842	0.56	0.636364
10	KevinMorrison	79	21	70	0.530201	0.79	0.634538
35	TheresePoletti	57	43	27	0.678571	0.57	0.619565
36	KeithWeir	63	37	41	0.605769	0.63	0.617647
16	KarlPenhaul	55	45	24	0.696203	0.55	0.614525
37	JoWinterbottom	50	50	13	0.793651	0.50	0.613497
23	DarrenSchuettler	53	47	31	0.630952	0.53	0.576087
30	JimGilchrist	47	53	17	0.734375	0.47	0.573171
40	MatthewBunce	62	38	57	0.521008	0.62	0.566210
39	JaneMacartney	52	48	38	0.577778	0.52	0.547368
17	MartinWolk	48	52	37	0.564706	0.48	0.518919
19	DavidLawder	39	61	15	0.722222	0.39	0.506494
31	BradDorfman	41	59	24	0.630769	0.41	0.496970
25	TanEeLyn	56	44	90	0.383562	0.56	0.455285
14	JanLopatka	42	58	54	0.437500	0.42	0.428571
9	EdnaFernandes	49	51	80	0.379845	0.49	0.427948
3	MichaelConnor	35	65	61	0.364583	0.35	0.357143
44	SarahDavison	27	73	49	0.355263	0.27	0.306818
26	PierreTran	16	84	17	0.484848	0.16	0.240602

Matrix Totals

Best Author: KevinDrawbaugh
Correct: 3339
Incorrect: 1661
Accuracy: 0.6678

Vectorizer Values: Stopwords = long, Stemming = False
KNN Values: K = 15, similarity measure = okapi

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
15	KevinDrawbaugh	94	6	8	0.921569	0.94	0.930693
42	PeterHumphrey	85	15	7	0.923913	0.85	0.885417
5	EricAuchard	91	9	20	0.819820	0.91	0.862559
4	JoeOrtiz	86	14	20	0.811321	0.86	0.834951
32	AlanCrosby	75	25	10	0.882353	0.75	0.810811
13	JohnMastrini	95	5	48	0.664336	0.95	0.781893
46	BernardHickey	99	1	58	0.630573	0.99	0.770428
20	FumikoFujisaki	84	16	39	0.682927	0.84	0.753363
34	BenjaminKangLim	88	12	46	0.656716	0.88	0.752137
1	LynnleyBrowning	86	14	44	0.661538	0.86	0.747826

11	SamuelPerry	73	27	23	0.760417	0.73	0.744898
48	AlexanderSmith	70	30	19	0.786517	0.70	0.740741
27	HeatherScoffield	83	17	44	0.653543	0.83	0.731278
2	KouroshKarimkhany	83	17	46	0.643411	0.83	0.724891
28	MureDickie	71	29	26	0.731959	0.71	0.720812
0	RobinSidel	70	30	26	0.729167	0.70	0.714286
8	LynneO'Donnell	59	41	8	0.880597	0.59	0.706587
29	RogerFillion	65	35	20	0.764706	0.65	0.702703
38	MarkBendeich	63	37	17	0.787500	0.63	0.700000
18	ScottHillis	64	36	20	0.761905	0.64	0.695652
33	JonathanBirt	77	23	45	0.631148	0.77	0.693694
7	SimonCowell	69	31	39	0.638889	0.69	0.663462
43	TimFarrand	56	44	17	0.767123	0.56	0.647399
45	GrahamEarnshaw	77	23	66	0.538462	0.77	0.633745
22	NickLouth	63	37	36	0.636364	0.63	0.633166
24	WilliamKazer	77	23	68	0.531034	0.77	0.628571
47	KirstinRidley	58	42	28	0.674419	0.58	0.623656
12	PatriciaCommins	52	48	15	0.776119	0.52	0.622754
19	DavidLawder	77	23	74	0.509934	0.77	0.613546
17	MartinWolk	82	18	90	0.476744	0.82	0.602941
41	ToddNissen	48	52	13	0.786885	0.48	0.596273
23	DarrenSchuettler	56	44	33	0.629213	0.56	0.592593
6	AaronPressman	59	41	42	0.584158	0.59	0.587065
30	JimGilchrist	45	55	11	0.803571	0.45	0.576923
40	MatthewBunce	55	45	36	0.604396	0.55	0.575916
35	TheresePoletti	44	56	9	0.830189	0.44	0.575163
36	KeithWeir	67	33	80	0.455782	0.67	0.542510
21	MarcelMichelson	45	55	25	0.642857	0.45	0.529412
39	JaneMacartney	51	49	43	0.542553	0.51	0.525773
37	JoWinterbottom	37	63	9	0.804348	0.37	0.506849
49	LydiaZajc	34	66	1	0.971429	0.34	0.503704
16	KarlPenhaul	50	50	49	0.505051	0.50	0.502513
31	BradDorfman	49	51	51	0.490000	0.49	0.490000
10	KevinMorrison	79	21	163	0.326446	0.79	0.461988
9	EdnaFernandes	48	52	75	0.390244	0.48	0.430493
14	JanLopatka	46	54	71	0.393162	0.46	0.423963
3	MichaelConnor	37	63	55	0.402174	0.37	0.385417
44	SarahDavison	19	81	36	0.345455	0.19	0.245161
26	PierreTran	15	85	13	0.535714	0.15	0.234375
25	TanEeLyn	1	99	1	0.500000	0.01	0.019608

Matrix Totals

Best Author: KevinDrawbaugh
Correct: 3157
Incorrect: 1843
Accuracy: 0.6314

Random Forest

```
Stopwords = long, Stemming = False, k=4000, N=10
```

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
28	LynnleyBrowning	92.0	8.0	6.0	0.938776	0.92	0.929293
26	LydiaZajc	88.0	12.0	2.0	0.977778	0.88	0.926316
42	SarahDavison	86.0	14.0	2.0	0.977273	0.86	0.914894
21	KeithWeir	88.0	12.0	5.0	0.946237	0.88	0.911917
6	DarrenSchuettler	86.0	14.0	3.0	0.966292	0.86	0.910053
32	MatthewBunce	85.0	15.0	2.0	0.977011	0.85	0.909091
7	DavidLawder	85.0	15.0	2.0	0.977011	0.85	0.909091
5	BradDorfman	89.0	11.0	7.0	0.927083	0.89	0.908163
23	KevinMorrison	85.0	15.0	3.0	0.965909	0.85	0.904255
12	HeatherScoffield	88.0	12.0	8.0	0.916667	0.88	0.897959
38	PierreTran	84.0	16.0	4.0	0.954545	0.84	0.893617
4	BernardHickey	82.0	18.0	2.0	0.976190	0.82	0.891304
18	JohnMastrini	88.0	12.0	11.0	0.888889	0.88	0.884422
22	KevinDrawbaugh	84.0	16.0	6.0	0.933333	0.84	0.884211
25	KouroshKarimkhany	80.0	20.0	2.0	0.975610	0.80	0.879121
16	JoWinterbottom	78.0	22.0	2.0	0.975000	0.78	0.866667
20	KarlPenhaul	78.0	22.0	2.0	0.975000	0.78	0.866667
48	ToddNissen	76.0	24.0	2.0	0.974359	0.76	0.853933
45	TanEeLyn	87.0	13.0	17.0	0.836538	0.87	0.852941
36	PatriciaCommins	77.0	23.0	5.0	0.939024	0.77	0.846154
39	RobinSidel	73.0	27.0	0.0	1.000000	0.73	0.843931
24	KirstinRidley	75.0	25.0	3.0	0.961538	0.75	0.842697
9	EricAuchard	74.0	26.0	2.0	0.973684	0.74	0.840909
10	FumikoFujisaki	72.0	28.0	0.0	1.000000	0.72	0.837209
17	JoeOrtiz	73.0	27.0	2.0	0.973333	0.73	0.834286
19	JonathanBirt	71.0	29.0	0.0	1.000000	0.71	0.830409
47	TimFarrand	72.0	28.0	3.0	0.960000	0.72	0.822857
34	MureDickie	90.0	10.0	29.0	0.756303	0.90	0.821918
29	MarcelMichelson	69.0	31.0	0.0	1.000000	0.69	0.816568
0	AaronPressman	69.0	31.0	0.0	1.000000	0.69	0.816568
30	MarkBendeich	69.0	31.0	1.0	0.985714	0.69	0.811765
37	PeterHumphrey	95.0	5.0	41.0	0.698529	0.95	0.805085
44	SimonCowell	69.0	31.0	3.0	0.958333	0.69	0.802326
1	AlanCrosby	75.0	25.0	12.0	0.862069	0.75	0.802139
27	LynneO'Donnell	86.0	14.0	29.0	0.747826	0.86	0.800000
46	ThesePoletti	64.0	36.0	0.0	1.000000	0.64	0.780488
2	AlexanderSmith	62.0	38.0	0.0	1.000000	0.62	0.765432
13	JanLopatka	62.0	38.0	0.0	1.000000	0.62	0.765432
3	BenjaminKangLim	61.0	39.0	1.0	0.983871	0.61	0.753086
33	MichaelConnor	58.0	42.0	0.0	1.000000	0.58	0.734177
43	ScottHillis	56.0	44.0	1.0	0.982456	0.56	0.713376

31	MartinWolk	88.0	12.0	61.0	0.590604	0.88	0.706827
11	GrahamEarnshaw	95.0	5.0	74.0	0.562130	0.95	0.706320
8	EdnaFernandes	55.0	45.0	2.0	0.964912	0.55	0.700637
15	JimGilchrist	98.0	2.0	83.0	0.541436	0.98	0.697509
14	JaneMacartney	51.0	49.0	1.0	0.980769	0.51	0.671053
49	WilliamKazer	47.0	53.0	0.0	1.000000	0.47	0.639456
40	RogerFillion	93.0	7.0	121.0	0.434579	0.93	0.592357
41	SamuelPerry	94.0	6.0	197.0	0.323024	0.94	0.480818
35	NickLouth	98.0	2.0	341.0	0.223235	0.98	0.363636

Matrix Totals

Best Author: LynnleyBrowning
Correct: 3900.0
Incorrect: 1100.0
Accuracy: 0.78
Time Taken: 4252.797071695328

Stopwords = onix, Stemming = True, k=2000, N=20

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
6	DarrenSchuettler	97.0	3.0	1.0	0.989796	0.97	0.979798
24	KirstinRidley	98.0	2.0	3.0	0.970297	0.98	0.975124
26	LydiaZajc	98.0	2.0	3.0	0.970297	0.98	0.975124
35	NickLouth	95.0	5.0	0.0	1.000000	0.95	0.974359
19	JonathanBirt	95.0	5.0	0.0	1.000000	0.95	0.974359
16	JoWinterbottom	94.0	6.0	1.0	0.989474	0.94	0.964103
10	FumikoFujisaki	94.0	6.0	1.0	0.989474	0.94	0.964103
29	MarcelMichelson	93.0	7.0	0.0	1.000000	0.93	0.963731
27	LynneO'Donnell	94.0	6.0	2.0	0.979167	0.94	0.959184
21	KeithWeir	93.0	7.0	1.0	0.989362	0.93	0.958763
31	MartinWolk	92.0	8.0	0.0	1.000000	0.92	0.958333
41	SamuelPerry	91.0	9.0	0.0	1.000000	0.91	0.952880
40	RogerFillion	90.0	10.0	0.0	1.000000	0.90	0.947368
23	KevinMorrison	90.0	10.0	0.0	1.000000	0.90	0.947368
20	KarlPenhaul	97.0	3.0	8.0	0.923810	0.97	0.946341
14	JaneMacartney	94.0	6.0	5.0	0.949495	0.94	0.944724
15	JimGilchrist	99.0	1.0	11.0	0.900000	0.99	0.942857
36	PatriciaCommins	90.0	10.0	1.0	0.989011	0.90	0.942408
2	AlexanderSmith	89.0	11.0	0.0	1.000000	0.89	0.941799
8	EdnaFernandes	90.0	10.0	2.0	0.978261	0.90	0.937500
47	TimFarrand	94.0	6.0	7.0	0.930693	0.94	0.935323
46	TheresePoletti	92.0	8.0	5.0	0.948454	0.92	0.934010
48	ToddNissen	89.0	11.0	2.0	0.978022	0.89	0.931937

17	JoeOrtiz	88.0	12.0	1.0	0.988764	0.88	0.931217
44	SimonCowell	100.0	0.0	15.0	0.869565	1.00	0.930233
37	PeterHumphrey	98.0	2.0	13.0	0.882883	0.98	0.928910
9	EricAuchard	87.0	13.0	1.0	0.988636	0.87	0.925532
0	AaronPressman	86.0	14.0	0.0	1.000000	0.86	0.924731
7	DavidLawder	88.0	12.0	3.0	0.967033	0.88	0.921466
11	GrahamEarnshaw	97.0	3.0	16.0	0.858407	0.97	0.910798
33	MichaelConnor	85.0	15.0	2.0	0.977011	0.85	0.909091
42	SarahDavison	84.0	16.0	1.0	0.988235	0.84	0.908108
22	KevinDrawbaugh	85.0	15.0	3.0	0.965909	0.85	0.904255
30	MarkBendeich	84.0	16.0	2.0	0.976744	0.84	0.903226
5	BradDorfman	88.0	12.0	7.0	0.926316	0.88	0.902564
12	HeatherScoffield	100.0	0.0	22.0	0.819672	1.00	0.900901
4	BernardHickey	97.0	3.0	20.0	0.829060	0.97	0.894009
3	BenjaminKangLim	83.0	17.0	3.0	0.965116	0.83	0.892473
49	WilliamKazer	81.0	19.0	2.0	0.975904	0.81	0.885246
39	RobinSidel	94.0	6.0	20.0	0.824561	0.94	0.878505
38	PierreTran	78.0	22.0	2.0	0.975000	0.78	0.866667
45	TanEeLyn	97.0	3.0	27.0	0.782258	0.97	0.866071
25	KouroshKarimkhany	100.0	0.0	36.0	0.735294	1.00	0.847458
34	MureDickie	74.0	26.0	1.0	0.986667	0.74	0.845714
32	MatthewBunce	99.0	1.0	40.0	0.712230	0.99	0.828452
28	LynnleyBrowning	68.0	32.0	0.0	1.000000	0.68	0.809524
43	ScottHillis	95.0	5.0	44.0	0.683453	0.95	0.794979
13	JanLopatka	97.0	3.0	62.0	0.610063	0.97	0.749035
1	AlanCrosby	95.0	5.0	86.0	0.524862	0.95	0.676157
18	JohnMastrini	42.0	58.0	0.0	1.000000	0.42	0.591549

Matrix Totals

Best Author: DarrenSchuettler
Correct: 4518.0
Incorrect: 482.0
Accuracy: 0.9036
Time Taken: 4529.934679031372

Stopwords = short, Stemming = False, k=4000, N=10

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
37	PeterHumphrey	96.0	4.0	0.0	1.000000	0.96	0.979592
6	DarrenSchuettler	94.0	6.0	1.0	0.989474	0.94	0.964103
26	LydiaZajc	98.0	2.0	8.0	0.924528	0.98	0.951456
41	SamuelPerry	79.0	21.0	1.0	0.987500	0.79	0.877778
27	LynneO'Donnell	95.0	5.0	28.0	0.772358	0.95	0.852018
25	KouroshKarimkhany	73.0	27.0	1.0	0.986486	0.73	0.839080
3	BenjaminKangLim	85.0	15.0	18.0	0.825243	0.85	0.837438
9	EricAuchard	70.0	30.0	0.0	1.000000	0.70	0.823529
7	DavidLawder	67.0	33.0	0.0	1.000000	0.67	0.802395
46	ThesePoletti	65.0	35.0	0.0	1.000000	0.65	0.787879
43	ScottHillis	62.0	38.0	0.0	1.000000	0.62	0.765432
44	SimonCowell	68.0	32.0	12.0	0.850000	0.68	0.755556
5	BradDorfman	60.0	40.0	1.0	0.983607	0.60	0.745342
35	NickLouth	58.0	42.0	0.0	1.000000	0.58	0.734177
49	WilliamKazer	58.0	42.0	0.0	1.000000	0.58	0.734177
42	SarahDavison	56.0	44.0	0.0	1.000000	0.56	0.717949
12	HeatherScoffield	51.0	49.0	0.0	1.000000	0.51	0.675497
22	KevinDrawbaugh	50.0	50.0	0.0	1.000000	0.50	0.666667
48	ToddNissen	44.0	56.0	0.0	1.000000	0.44	0.611111
31	MartinWolk	43.0	57.0	0.0	1.000000	0.43	0.601399
1	AlanCrosby	43.0	57.0	0.0	1.000000	0.43	0.601399
45	TanEeLyn	41.0	59.0	0.0	1.000000	0.41	0.581560
21	KeithWeir	41.0	59.0	0.0	1.000000	0.41	0.581560
36	PatriciaCommins	41.0	59.0	0.0	1.000000	0.41	0.581560
15	JimGilchrist	39.0	61.0	0.0	1.000000	0.39	0.561151
16	JoWinterbottom	39.0	61.0	0.0	1.000000	0.39	0.561151
10	FumikoFujisaki	38.0	62.0	0.0	1.000000	0.38	0.550725
28	LynnleyBrowning	38.0	62.0	0.0	1.000000	0.38	0.550725
29	MarcelMichelson	38.0	62.0	0.0	1.000000	0.38	0.550725
39	RobinSidel	36.0	64.0	0.0	1.000000	0.36	0.529412
33	MichaelConnor	35.0	65.0	0.0	1.000000	0.35	0.518519
0	AaronPressman	34.0	66.0	0.0	1.000000	0.34	0.507463
24	KirstinRidley	33.0	67.0	0.0	1.000000	0.33	0.496241
23	KevinMorrison	33.0	67.0	0.0	1.000000	0.33	0.496241
40	RogerFillion	32.0	68.0	0.0	1.000000	0.32	0.484848
18	JohnMastrini	32.0	68.0	0.0	1.000000	0.32	0.484848
32	MatthewBunce	31.0	69.0	0.0	1.000000	0.31	0.473282
8	EdnaFernandes	30.0	70.0	0.0	1.000000	0.30	0.461538
19	JonathanBirt	74.0	26.0	184.0	0.286822	0.74	0.413408
47	TimFarrand	24.0	76.0	0.0	1.000000	0.24	0.387097
30	MarkBendeich	24.0	76.0	0.0	1.000000	0.24	0.387097
14	JaneMacartney	23.0	77.0	0.0	1.000000	0.23	0.373984
2	AlexanderSmith	23.0	77.0	0.0	1.000000	0.23	0.373984
34	MureDickie	17.0	83.0	0.0	1.000000	0.17	0.290598
13	JanLopatka	16.0	84.0	0.0	1.000000	0.16	0.275862

11	GrahamEarnshaw	15.0	85.0	0.0	1.000000	0.15	0.260870
38	PierreTran	10.0	90.0	0.0	1.000000	0.10	0.181818
17	JoeOrtiz	10.0	90.0	0.0	1.000000	0.10	0.181818
20	KarlPenhaul	8.0	92.0	0.0	1.000000	0.08	0.148148
4	BernardHickey	98.0	2.0	2378.0	0.039580	0.98	0.076087

Matrix Totals

Best Author: PeterHumphrey
Correct: 2368.0
Incorrect: 2632.0
Accuracy: 0.4736
Time Taken: 7038.2494831085205

Stopwords = long, Stemming = True, k=4000, N=10

	Author	Hits	Misses	Strikes	Precision	Recall	F-Measure
15	JimGilchrist	100.0	0.0	1.0	0.990099	1.00	0.995025
23	KevinMorrison	97.0	3.0	0.0	1.000000	0.97	0.984772
25	KouroshKarimkhany	96.0	4.0	0.0	1.000000	0.96	0.979592
40	RogerFillion	96.0	4.0	0.0	1.000000	0.96	0.979592
27	LynneO'Donnell	96.0	4.0	0.0	1.000000	0.96	0.979592
26	LydiaZajc	96.0	4.0	0.0	1.000000	0.96	0.979592
45	TanEeLyn	97.0	3.0	2.0	0.979798	0.97	0.974874
3	BenjaminKangLim	97.0	3.0	2.0	0.979798	0.97	0.974874
6	DarrenSchuettler	96.0	4.0	1.0	0.989691	0.96	0.974619
7	DavidLawder	96.0	4.0	1.0	0.989691	0.96	0.974619
35	NickLouth	95.0	5.0	0.0	1.000000	0.95	0.974359
14	JaneMacartney	100.0	0.0	6.0	0.943396	1.00	0.970874
0	AaronPressman	95.0	5.0	1.0	0.989583	0.95	0.969388
36	PatriciaCommins	94.0	6.0	0.0	1.000000	0.94	0.969072
10	FumikoFujisaki	94.0	6.0	0.0	1.000000	0.94	0.969072
29	MarcelMichelson	94.0	6.0	1.0	0.989474	0.94	0.964103
13	JanLopatka	98.0	2.0	7.0	0.933333	0.98	0.956098
18	JohnMastrini	91.0	9.0	0.0	1.000000	0.91	0.952880
31	MartinWolk	91.0	9.0	0.0	1.000000	0.91	0.952880
34	MureDickie	90.0	10.0	0.0	1.000000	0.90	0.947368
49	WilliamKazer	90.0	10.0	0.0	1.000000	0.90	0.947368
9	EricAuchard	89.0	11.0	0.0	1.000000	0.89	0.941799
43	ScottHillis	89.0	11.0	0.0	1.000000	0.89	0.941799
41	SamuelPerry	89.0	11.0	0.0	1.000000	0.89	0.941799
20	KarlPenhaul	89.0	11.0	0.0	1.000000	0.89	0.941799

28	LynnleyBrowning	100.0	0.0	13.0	0.884956	1.00	0.938967
11	GrahamEarnshaw	89.0	11.0	1.0	0.988889	0.89	0.936842
1	AlanCrosby	89.0	11.0	1.0	0.988889	0.89	0.936842
46	ThieresePoletti	95.0	5.0	8.0	0.922330	0.95	0.935961
22	KevinDrawbaugh	88.0	12.0	1.0	0.988764	0.88	0.931217
42	SarahDavison	87.0	13.0	0.0	1.000000	0.87	0.930481
24	KirstinRidley	87.0	13.0	0.0	1.000000	0.87	0.930481
37	PeterHumphrey	100.0	0.0	15.0	0.869565	1.00	0.930233
33	MichaelConnor	94.0	6.0	9.0	0.912621	0.94	0.926108
39	RobinSidel	87.0	13.0	1.0	0.988636	0.87	0.925532
48	ToddNissen	85.0	15.0	0.0	1.000000	0.85	0.918919
17	JoeOrtiz	84.0	16.0	0.0	1.000000	0.84	0.913043
12	HeatherScoffield	83.0	17.0	0.0	1.000000	0.83	0.907104
2	AlexanderSmith	82.0	18.0	0.0	1.000000	0.82	0.901099
32	MatthewBunce	81.0	19.0	0.0	1.000000	0.81	0.895028
44	SimonCowell	80.0	20.0	0.0	1.000000	0.80	0.888889
38	PierreTran	75.0	25.0	0.0	1.000000	0.75	0.857143
30	MarkBendeich	75.0	25.0	0.0	1.000000	0.75	0.857143
16	JoWinterbottom	67.0	33.0	0.0	1.000000	0.67	0.802395
21	KeithWeir	66.0	34.0	0.0	1.000000	0.66	0.795181
8	EdnaFernandes	54.0	46.0	0.0	1.000000	0.54	0.701299
19	JonathanBirt	53.0	47.0	0.0	1.000000	0.53	0.692810
47	TimFarrand	51.0	49.0	1.0	0.980769	0.51	0.671053
5	BradDorfman	96.0	4.0	104.0	0.480000	0.96	0.640000
4	BernardHickey	100.0	0.0	421.0	0.191939	1.00	0.322061

Matrix Totals

Best Author: JimGilchrist
Correct: 4403.0
Incorrect: 597.0
Accuracy: 0.8806
Time Taken: 8248.204617977142