Automatic Tweet Hashtag Categorization

Daniel Schultz and Sundeep Jolly

December 9, 2010

1 Introduction

Twitter is a popular micro-blogging service allowing users to publish short messages (i.e., limited to 140 characters) to a network of contacts and the entire web community, if desired. These short messages "tweets") often convey information or opinions that are readily associated with a given topic (e.g., current event, new movie, or other trends). "Hashtags" allow users to self-categorize their tweets using a pound symbol (#) followed by a word or phrase indicating the categorization. For example, the tweet "*Body scanner makers doubled lobbying money over five years #TSA*" (in reference to a recent TSA policy change) is hash-tagged with #TSA to indicate its content and suitable categorization. Hashtags have emerged as being useful for grouping tweets of similar content, sentiment, or nature regarding a given topic.

Despite the ubiquity and utility of hashtags for categorization and grouping, a vast number of tweets related to a given categorization go untagged at publication. Furthermore, tagged tweets related to a categorization are often not tagged with the same hashtag (e.g., #FireOfLondon vs. #LondonFire vs. #LDNFire), leading to confusion, inefficient grouping, and lack of consistency.

In order to promote consistency amongst popular hashtags (and thereby enable and promote hashtag-based social networking on Twitter), a capability for automatic tweet hashtag characterization is desirable. This project aims to examine the utility of several machine learning techniques in conjunction with appropriate feature selection algorithms for the text categorization problem posed by automatic hashtagging.

2 Data Collection and Preprocessing

2.1 Data Collection

To obtain a representative dataset the public Twitter API was scraped directly for English tweets over a two-week period. Tweets were stripped of usernames, urls, and retweet indicators (RT) to protect against noise and duplicates. Each additional datapoint was codified and compared with all existing points in order to prevent duplicate entries. The tags were selected through Twitters list of global trending tags.

The final dataset consisted of 24 tags with 1500 training points (100 of which were used for validation) and 500 testing points per tag, resulting in a pool of 36,000 training points and 12,000 test points. Two random subsets of size 5 and 10 were selected from the full tag pool in order to test how algorithms performed on the dataset as the number of categories increased.

2.2 Preprocessing

Before training the models, the tweets were converted to vectors using the Bag of Words algorithm. The resulting vectors contained a huge number of dimensions, indicating a need for dimensionality reduction. The dataset was run through PCA as well as a simple variance based reduction metric. To explore these methods, each case was run using four treatments: 10% dictionary size, 50% dictionary size, 100% dictionary size, and 90% energy PCA.

3 Methods

Studies in the literature report on the wide use of the bag-of-words feature model in conjunction with machine learning techniques for the text categorization problem [**?**, **?**, **?**]. To approach the categorization problem, Naïve Bayes, k-Nearest Neighbor, and Support Vector Machine algorithms were employed. All algorithms were implemented in MATLAB.

3.1 Naïve Bayes Classifier

The Naïve Bayes classifier is a token-based classifier (i.e., operates on words as features directly rather than on projected principal components or other eigenprojections of words). Specifically, Naïve Bayes employs the bag-of-words assumption to calculate tweet likelihoods and hashtag posterior probabilities, classifying a tweet with the hashtag that maximizes the posterior probability.

The classifier requires that the probability of any word w in the total dictionary afforded by the entire training set given any hashtag H (i.e., $p(w|H)$) in the training set be calculated and stored in a look-up table for use in the likelihood calculations. For the current implementation, $p(w|H)$ is defined as the ratio of the number of occurrences of the word w within all tweets in the hashtag H to the number of occurrences of the word w within the *entire* training set.

For a tweet T to be classified, the likelihood $p(T|H)$ of the tweet given a hashtag H is computed as the product of the probabilities of its constitutent words (tokens) given that hashtag as:

$$p\left(T|H\right) = \prod_{i=1}^{N} p\left(w_i|H\right) \tag{1}$$

where N is the number of words in tweet T. Note that for the current implementation, if a word w_i is not present in the training set dictionary, it is ignored in the likelihood calculation.

The final step in the classification algorithm involves the calculation of the posterior probability of the hashtag being true given the tweet, $p(H|T)$, using Bayes' theorem as $p(H|T) = \frac{p(T|H)p(H)}{p(T)}$, where $p(H)$ is the prior probability of the hashtag H (defined to be uniform over all hashtags) and $p(T)$ is a normalization factor. The classification rule is then $\arg\max_H p(H|T) = \arg\max_H p(T|H)$.

3.2 *k*-Nearest Neighbor Classifier

The K-Nearest Neighbor algorithm performs classification estimates by assigning test data samples to the labels with the most similar training data. KNN was run for K=1-100 for the 5-category, 10-category, and full-category cases. Unfortunately, due to computational limitations several of the larger-set cases were not able to complete in time for this paper.

3.3 Support Vector Machine Classifier

The support vector machine classifier is popular within the domain of text categorization. Studies in the literature [**?**, **?**] report on the use of various SVM kernels for the text categorization problem; consistent with previous approaches, the current implementation uses the linear, quadratic, and radial basis function kernels. For the quadratic and radial kernels, the factos γ was chosen to be $1/k$, where k is the number of features.

The implementation of the SVM classifier used MATLAB in conjunction with the freely-available LIBSVM library[1].

4 Results and Analysis

Detailed results for all classifiers, over all training sets and testing sets used, and for all dictionary sizes (or alternatively, PCA) are included in the appendix.

After analyzing the validation results for the smaller dictionary sizes, it was consistently clear that smaller dictionary sizes using a simple variance metric removed too many essential features and resulted in data points, which shared no common words with the training sets. After accounting for uncategorized data points, PCA was by far the most effective dimensionality reduction technique for KNN and SVM.

Of the methods tested, SVM using a linear kernel provided the highest amount of accuracy, while SVM with a quadratic kernel yielded the worst. PCA was the most effective method for dimensionality reduction, yielding the best accuracy to computation time ratio. KNN and Bayes performed decently, but not as well as the linear kernel SVM. In general there was a decrease in accuracy as the number of categories increased, but the accuracy to category count ratio increased (i.e. as categories were added, the good-performing algorithms tended to do increasingly better than chance).

[1]downloaded from http://www.csie.ntu.edu.tw/ cjlin/libsvm/, courtesy of Chih-Chung Chang and Chih-Jen Lin, National Taiwan University

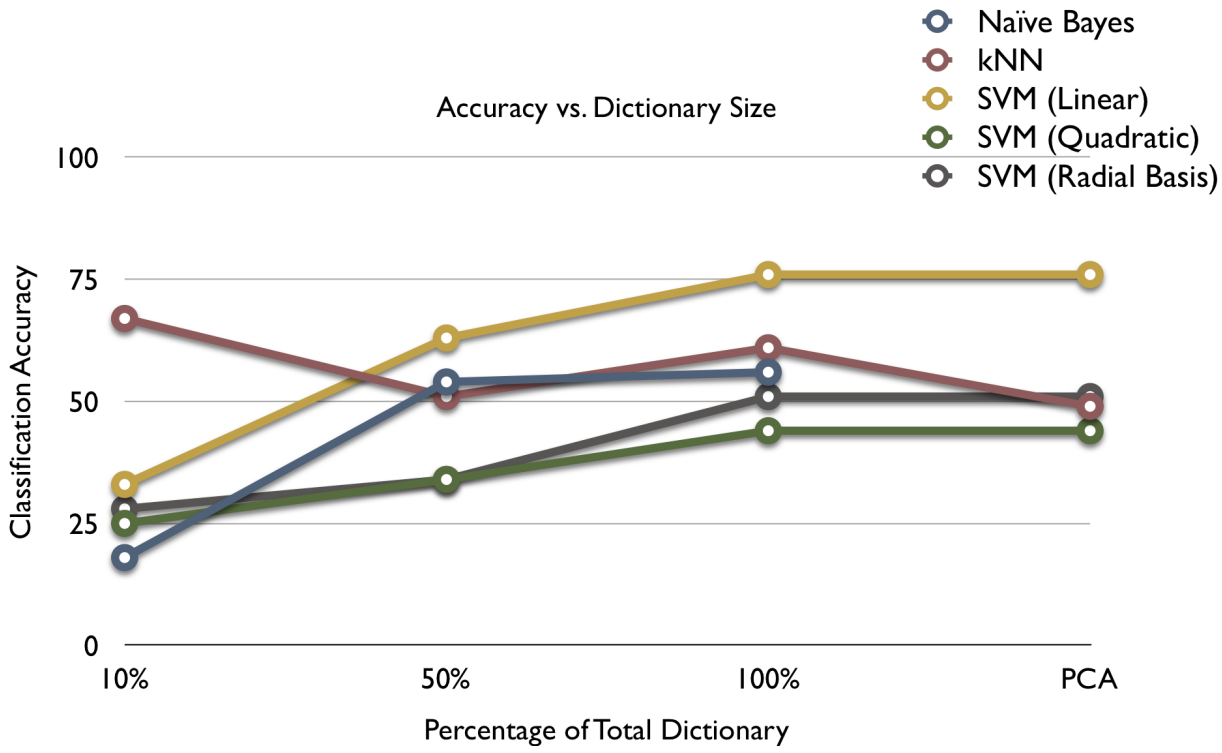


Figure 1: Comparative accuracies of classifiers for 5 hashtags.

5 Future Work

There are several obvious ways for this exploration to be expanded:

1. Incorporating more tags and increasing the number of training samples would augment the relevance and of the categorization algorithm.
2. Performing data cleanup methods in order to correct spelling variations and typos would be particularly useful on a data set like Twitters, where shortened words and poor grammar are incredibly common.
3. Applying other categorization algorithms such as Artificial Neural Networks or Decision Trees.
4. Utilizing alternate textual feature selection and dimensionality reduction techniques.

References

[1] Yang. An evaluation of statistical approaches to text categorization. *Information retrieval* (1999) vol. 1 (1) pp. 69-90

[2] Yang and Chute. A linear least squares fit mapping method for information retrieval from natural language texts. *Proceedings of the 14th conference on Computational linguistics-Volume 2* (1992) pp. 447-453

[3] Kwok. Automated Text Categorization Using Support Vector Machine. In *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING* (ICONIP (1998)) pp. 347–351

Appendix: Results

kNN, 5 labels, PCA dimensionality reduction

k	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
1	0.4878	0.3273	0.5030	0.5130	0.3972	0.6986
2	0.5038	0.4691	0.5230	0.5289	0.3852	0.6128
3	0.5154	0.4451	0.5190	0.5449	0.3593	0.7086
4	0.5158	0.4152	0.5030	0.5589	0.4012	0.7006
5	0.5226	0.3912	0.5150	0.5629	0.4052	0.7385
6	0.5198	0.3852	0.5150	0.5669	0.4052	0.7265
7	0.5253	0.3792	0.4910	0.5589	0.4571	0.7405
8	0.5226	0.3752	0.5030	0.5409	0.4651	0.7285
9	0.5269	0.3872	0.5010	0.5529	0.4531	0.7405
10	0.5174	0.3653	0.4910	0.5549	0.4371	0.7385
11	0.5174	0.3593	0.5090	0.5250	0.4551	0.7385
12	0.5138	0.3573	0.5070	0.5210	0.4531	0.7305
13	0.5098	0.3613	0.4950	0.5030	0.4511	0.7385
14	0.5070	0.3613	0.4850	0.4790	0.4790	0.7305
15	0.5110	0.3693	0.4770	0.4850	0.4790	0.7445
16	0.5078	0.3812	0.4671	0.4731	0.4711	0.7465
17	0.5074	0.3912	0.4611	0.4631	0.4810	0.7405
18	0.5090	0.3992	0.4511	0.4671	0.4830	0.7445
19	0.5062	0.4092	0.4491	0.4651	0.4691	0.7385
20	0.4978	0.4132	0.4431	0.4411	0.4631	0.7285
21	0.4946	0.4152	0.4291	0.4371	0.4611	0.7305
22	0.4906	0.4351	0.4251	0.4271	0.4391	0.7265
23	0.4958	0.4471	0.4311	0.4232	0.4471	0.7305
24	0.4858	0.4591	0.4212	0.4092	0.4251	0.7146
25	0.4858	0.4671	0.4331	0.3952	0.4232	0.7106
26	0.4854	0.4870	0.4391	0.3872	0.4092	0.7046
27	0.4814	0.4930	0.4331	0.3772	0.4032	0.7006
28	0.4814	0.5110	0.4351	0.3653	0.4012	0.6946
29	0.4778	0.5230	0.4391	0.3553	0.3772	0.6946
30	0.4770	0.5289	0.4311	0.3473	0.3792	0.6986
31	0.4782	0.5269	0.4331	0.3313	0.4072	0.6926
32	0.4723	0.5250	0.4311	0.3194	0.4012	0.6846
33	0.4711	0.5230	0.4351	0.3174	0.3972	0.6826
34	0.4699	0.5190	0.4391	0.3094	0.4012	0.6806

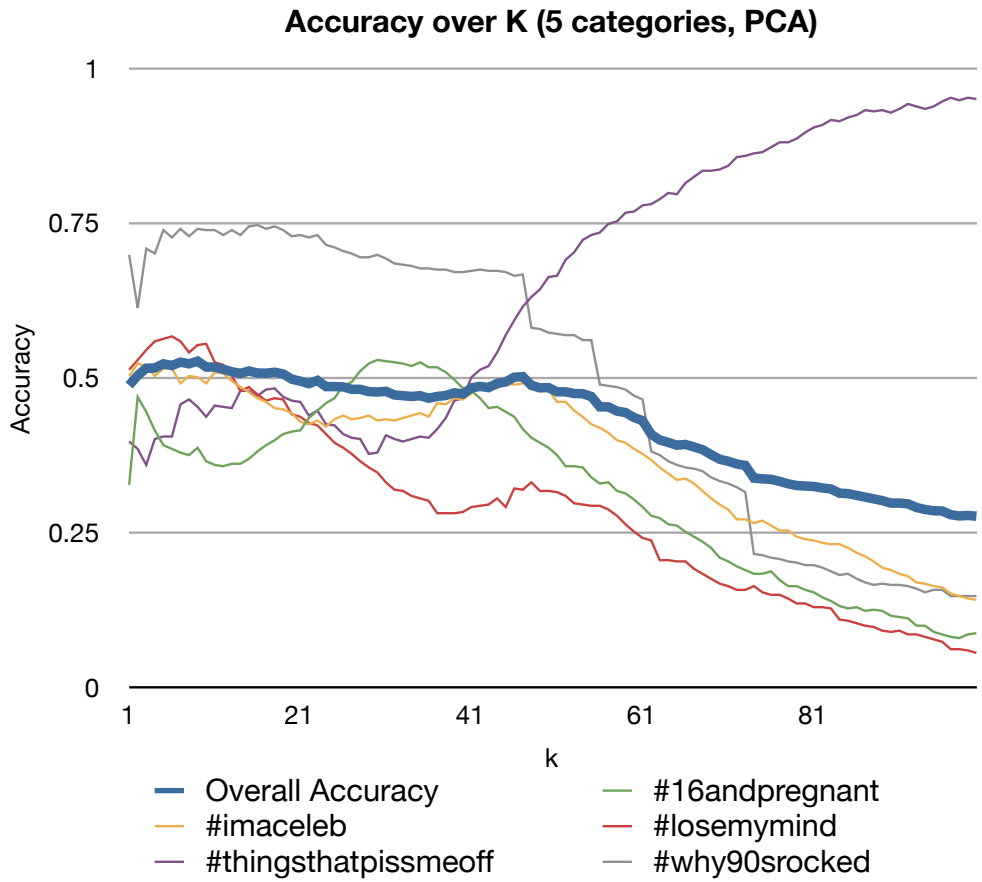
kNN, 5 labels, PCA dimensionality reduction

k	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
35	0.4711	0.5250	0.4431	0.3054	0.4052	0.6766
36	0.4671	0.5170	0.4371	0.3014	0.4032	0.6766
37	0.4699	0.5170	0.4591	0.2814	0.4172	0.6747
38	0.4715	0.5090	0.4571	0.2814	0.4351	0.6747
39	0.4758	0.4970	0.4671	0.2814	0.4631	0.6707
40	0.4739	0.4830	0.4651	0.2834	0.4671	0.6707
41	0.4838	0.4790	0.4750	0.2914	0.5010	0.6727
42	0.4862	0.4671	0.4830	0.2934	0.5130	0.6747
43	0.4842	0.4511	0.4830	0.2954	0.5190	0.6727
44	0.4910	0.4531	0.4830	0.3054	0.5409	0.6727
45	0.4934	0.4451	0.4910	0.2914	0.5689	0.6707
46	0.5010	0.4371	0.4890	0.3214	0.5928	0.6647
47	0.5018	0.4172	0.4910	0.3194	0.6148	0.6667
48	0.4878	0.4032	0.4930	0.3313	0.6307	0.5808
49	0.4838	0.3952	0.4850	0.3174	0.6427	0.5788
50	0.4842	0.3872	0.4810	0.3174	0.6627	0.5729
51	0.4774	0.3752	0.4611	0.3154	0.6647	0.5709
52	0.4774	0.3573	0.4611	0.3094	0.6906	0.5689
53	0.4747	0.3573	0.4471	0.2974	0.7026	0.5689
54	0.4743	0.3553	0.4371	0.2954	0.7226	0.5609
55	0.4699	0.3393	0.4251	0.2934	0.7305	0.5609
56	0.4531	0.3293	0.4192	0.2934	0.7345	0.4890
57	0.4531	0.3313	0.4112	0.2874	0.7485	0.4870
58	0.4463	0.3174	0.3992	0.2774	0.7525	0.4850
59	0.4439	0.3134	0.3952	0.2635	0.7665	0.4810
60	0.4359	0.3034	0.3852	0.2515	0.7685	0.4711
61	0.4307	0.2914	0.3772	0.2415	0.7784	0.4651
62	0.4088	0.2774	0.3673	0.2375	0.7804	0.3812
63	0.3996	0.2735	0.3553	0.2056	0.7884	0.3752
64	0.3956	0.2635	0.3453	0.2056	0.7984	0.3653
65	0.3912	0.2615	0.3353	0.2036	0.7964	0.3593
66	0.3924	0.2515	0.3373	0.2036	0.8144	0.3553
67	0.3884	0.2435	0.3293	0.1916	0.8244	0.3533
68	0.3840	0.2355	0.3174	0.1836	0.8343	0.3493

kNN, 5 labels, PCA dimensionality reduction

k	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
69	0.3760	0.2255	0.3054	0.1756	0.8343	0.3393
70	0.3685	0.2096	0.2954	0.1677	0.8363	0.3333
71	0.3653	0.2036	0.2874	0.1637	0.8423	0.3293
72	0.3609	0.1956	0.2715	0.1577	0.8563	0.3234
73	0.3585	0.1896	0.2715	0.1577	0.8583	0.3154
74	0.3381	0.1836	0.2655	0.1637	0.8623	0.2156
75	0.3369	0.1836	0.2695	0.1537	0.8643	0.2136
76	0.3361	0.1876	0.2615	0.1497	0.8723	0.2096
77	0.3329	0.1737	0.2535	0.1497	0.8802	0.2076
78	0.3289	0.1637	0.2535	0.1437	0.8802	0.2036
79	0.3261	0.1637	0.2435	0.1357	0.8862	0.2016
80	0.3253	0.1577	0.2395	0.1357	0.8962	0.1976
81	0.3246	0.1537	0.2375	0.1297	0.9042	0.1976
82	0.3222	0.1457	0.2335	0.1297	0.9082	0.1936
83	0.3206	0.1397	0.2315	0.1277	0.9162	0.1876
84	0.3138	0.1317	0.2315	0.1098	0.9142	0.1816
85	0.3130	0.1277	0.2255	0.1078	0.9202	0.1836
86	0.3102	0.1297	0.2176	0.1038	0.9242	0.1756
87	0.3074	0.1238	0.2116	0.0998	0.9321	0.1697
88	0.3046	0.1257	0.2036	0.0978	0.9301	0.1657
89	0.3018	0.1238	0.1936	0.0918	0.9321	0.1677
90	0.2978	0.1158	0.1896	0.0898	0.9281	0.1657
91	0.2978	0.1138	0.1836	0.0918	0.9341	0.1657
92	0.2966	0.1118	0.1796	0.0858	0.9421	0.1637
93	0.2906	0.0998	0.1697	0.0858	0.9381	0.1597
94	0.2874	0.0998	0.1677	0.0818	0.9341	0.1537
95	0.2854	0.0898	0.1637	0.0778	0.9381	0.1577
96	0.2850	0.0858	0.1617	0.0739	0.9461	0.1577
97	0.2790	0.0818	0.1517	0.0619	0.9521	0.1477
98	0.2770	0.0798	0.1477	0.0619	0.9481	0.1477
99	0.2778	0.0858	0.1437	0.0599	0.9521	0.1477
100	0.2766	0.0878	0.1417	0.0559	0.9501	0.1477

kNN, 5 labels, PCA dimensionality reduction



kNN, 5 labels, 100% dictionary size (no dimensionality reduction)

k	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
1	0.6088	0.8263	0.5824	0.7405	0.2512	0.6096
2	0.5550	0.9343	0.4910	0.6385	0.1154	0.5545
3	0.6042	0.9280	0.5556	0.7197	0.1671	0.6135
4	0.5702	0.9487	0.5142	0.6010	0.1401	0.6114
5	0.5838	0.9378	0.5693	0.5907	0.1471	0.6218
6	0.5723	0.9523	0.5348	0.5649	0.1299	0.6269
7	0.5949	0.9461	0.5556	0.6286	0.1487	0.6310
8	0.5849	0.9557	0.5377	0.5884	0.1608	0.6253
9	0.5975	0.9575	0.5528	0.6069	0.1759	0.6401
10	0.6007	0.9600	0.5621	0.6059	0.1633	0.6490
11	0.5971	0.9561	0.5629	0.5921	0.1684	0.6418
12	0.5902	0.9489	0.5549	0.5535	0.1825	0.6501
13	0.5936	0.9450	0.5595	0.5605	0.1871	0.6528
14	0.5809	0.9472	0.5529	0.5413	0.1491	0.6483
15	0.5946	0.9528	0.5684	0.5646	0.1593	0.6604
16	0.5937	0.9551	0.5611	0.5709	0.1515	0.6603
17	0.6033	0.9572	0.5924	0.5775	0.1544	0.6656
18	0.5989	0.9626	0.5877	0.5461	0.1647	0.6700
19	0.5956	0.9552	0.6034	0.5269	0.1554	0.6678
20	0.5931	0.9474	0.6000	0.5261	0.1510	0.6667
21	0.5956	0.9429	0.6092	0.5344	0.1488	0.6700
22	0.5958	0.9424	0.5972	0.5311	0.1435	0.6849
23	0.5915	0.9265	0.5914	0.5195	0.1545	0.6877
24	0.5882	0.9248	0.5799	0.5179	0.1595	0.6866
25	0.5242	0.9170	0.5741	0.5023	0.1542	0.2960
26	0.5298	0.9240	0.5736	0.5048	0.1577	0.3058
27	0.5276	0.9294	0.5731	0.5073	0.1493	0.3051
28	0.5260	0.9280	0.5668	0.4923	0.1521	0.3214
29	0.5313	0.9309	0.5542	0.5000	0.1636	0.3604
30	0.5342	0.9259	0.5546	0.5131	0.1651	0.3727
31	0.5277	0.9170	0.5478	0.5054	0.1627	0.3611
32	0.5275	0.9038	0.5482	0.5054	0.1650	0.3774

kNN, 5 labels, 100% dictionary size (no dimensionality reduction)

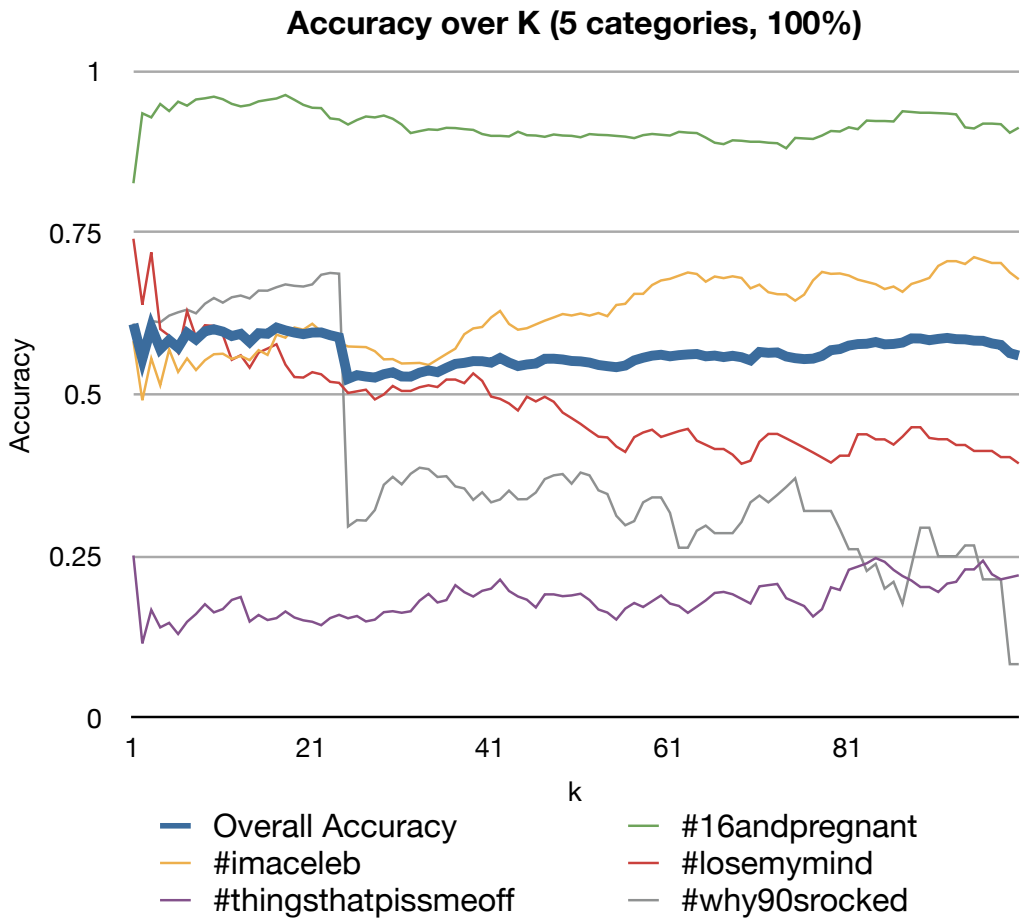
k	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
33	0.5336	0.9068	0.5487	0.5111	0.1814	0.3868
34	0.5371	0.9095	0.5455	0.5141	0.1919	0.3846
35	0.5344	0.9087	0.5540	0.5114	0.1786	0.3725
36	0.5411	0.9119	0.5625	0.5230	0.1823	0.3737
37	0.5472	0.9115	0.5707	0.5230	0.2053	0.3579
38	0.5488	0.9099	0.5930	0.5174	0.1946	0.3548
39	0.5515	0.9087	0.6020	0.5325	0.1878	0.3371
40	0.5513	0.9023	0.6042	0.5210	0.1966	0.3488
41	0.5491	0.8995	0.6190	0.4967	0.2000	0.3333
42	0.5563	0.8995	0.6290	0.4932	0.2139	0.3377
43	0.5492	0.8986	0.6089	0.4861	0.1964	0.3514
44	0.5438	0.9059	0.6000	0.4752	0.1879	0.3380
45	0.5463	0.9005	0.6023	0.4965	0.1829	0.3382
46	0.5477	0.9000	0.6082	0.4892	0.1709	0.3485
47	0.5551	0.8980	0.6140	0.4964	0.1911	0.3692
48	0.5552	0.9016	0.6190	0.4887	0.1911	0.3750
49	0.5540	0.9000	0.6242	0.4720	0.1883	0.3770
50	0.5517	0.8995	0.6220	0.4634	0.1895	0.3621
51	0.5511	0.8973	0.6250	0.4545	0.1921	0.3793
52	0.5492	0.9022	0.6218	0.4444	0.1824	0.3750
53	0.5455	0.9011	0.6258	0.4348	0.1678	0.3519
54	0.5437	0.9006	0.6209	0.4336	0.1631	0.3462
55	0.5421	0.8994	0.6382	0.4196	0.1522	0.3125
56	0.5446	0.8983	0.6400	0.4112	0.1691	0.2979
57	0.5526	0.8960	0.6554	0.4340	0.1778	0.3043
58	0.5567	0.9006	0.6554	0.4412	0.1716	0.3333
59	0.5599	0.9024	0.6690	0.4455	0.1805	0.3409
60	0.5613	0.9012	0.6761	0.4343	0.1894	0.3409
61	0.5589	0.9000	0.6786	0.4388	0.1769	0.3171
62	0.5607	0.9057	0.6835	0.4433	0.1732	0.2632
63	0.5618	0.9045	0.6884	0.4468	0.1626	0.2632
64	0.5625	0.9038	0.6861	0.4286	0.1721	0.2895

kNN, 5 labels, 100% dictionary size (no dimensionality reduction)

k	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
65	0.5589	0.8968	0.6742	0.4222	0.1818	0.2973
66	0.5600	0.8889	0.6822	0.4157	0.1933	0.2857
67	0.5577	0.8867	0.6797	0.4157	0.1949	0.2857
68	0.5597	0.8926	0.6825	0.4070	0.1913	0.2857
69	0.5575	0.8919	0.6800	0.3929	0.1842	0.3030
70	0.5526	0.8904	0.6639	0.3976	0.1770	0.3333
71	0.5656	0.8904	0.6696	0.4268	0.2035	0.3438
72	0.5643	0.8889	0.6579	0.4390	0.2054	0.3333
73	0.5649	0.8881	0.6549	0.4390	0.2072	0.3448
74	0.5586	0.8803	0.6545	0.4321	0.1852	0.3571
75	0.5560	0.8963	0.6449	0.4250	0.1792	0.3704
76	0.5546	0.8955	0.6542	0.4177	0.1731	0.3200
77	0.5553	0.8947	0.6762	0.4103	0.1569	0.3200
78	0.5596	0.9000	0.6893	0.4026	0.1683	0.3200
79	0.5684	0.9070	0.6863	0.3947	0.2020	0.3200
80	0.5701	0.9062	0.6869	0.4054	0.1979	0.2917
81	0.5755	0.9127	0.6837	0.4054	0.2292	0.2609
82	0.5778	0.9098	0.6774	0.4384	0.2340	0.2609
83	0.5783	0.9231	0.6739	0.4384	0.2391	0.2273
84	0.5810	0.9224	0.6703	0.4306	0.2472	0.2381
85	0.5774	0.9224	0.6628	0.4306	0.2414	0.2000
86	0.5780	0.9217	0.6667	0.4225	0.2289	0.2105
87	0.5801	0.9375	0.6585	0.4348	0.2195	0.1765
88	0.5866	0.9364	0.6707	0.4493	0.2125	0.2353
89	0.5864	0.9352	0.6750	0.4493	0.2025	0.2941
90	0.5838	0.9352	0.6800	0.4328	0.2025	0.2941
91	0.5858	0.9346	0.6986	0.4308	0.1948	0.2500
92	0.5873	0.9340	0.7059	0.4308	0.2078	0.2500
93	0.5854	0.9327	0.7059	0.4219	0.2105	0.2500
94	0.5851	0.9126	0.7015	0.4219	0.2297	0.2667
95	0.5831	0.9109	0.7121	0.4127	0.2297	0.2667
96	0.5828	0.9184	0.7077	0.4127	0.2432	0.2143

kNN, 5 labels, 100% dictionary size (no dimensionality reduction)

k	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
97	0.5788	0.9184	0.7031	0.4127	0.2222	0.2143
98	0.5765	0.9175	0.7031	0.4032	0.2143	0.2143
99	0.5638	0.9043	0.6885	0.4032	0.2174	0.0833
100	0.5601	0.9121	0.6780	0.3934	0.2206	0.0833



kNN, 10 labels, PCA dimensionality reduction

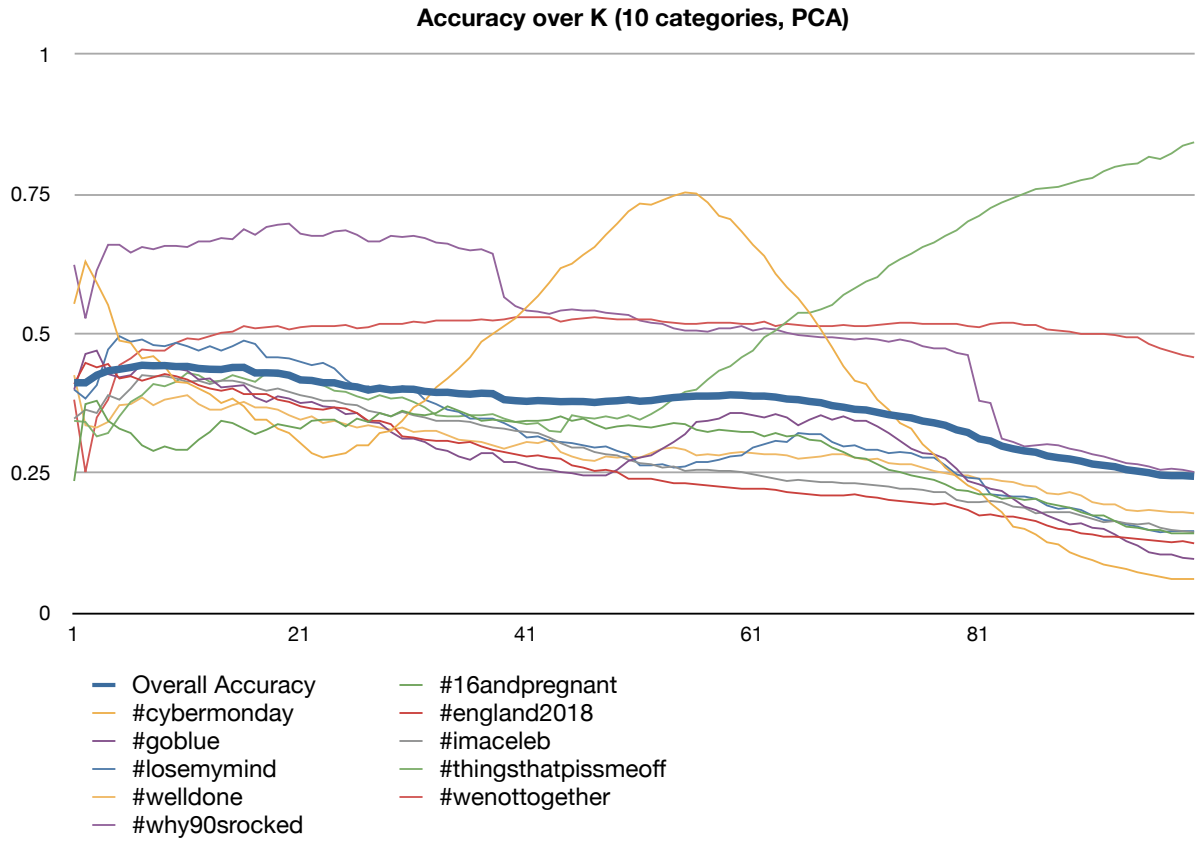
	Overall Accuracy	#16andpregnant	#cybermonday	#england2018	#goblue	#imaceleb	#losemymind	#thingsthatpissmeoff	#welldone	#wenottogether	#why90srocked
1	0.4116	0.2355	0.5529	0.4092	0.3992	0.3473	0.3992	0.3433	0.4251	0.3812	0.6228
2	0.4114	0.3733	0.6287	0.4471	0.4631	0.3633	0.3832	0.3413	0.3353	0.2515	0.5269
3	0.4251	0.3792	0.5908	0.4391	0.4691	0.3573	0.4072	0.3154	0.3313	0.3493	0.6128
4	0.4329	0.3433	0.5509	0.4451	0.4271	0.3892	0.4711	0.3214	0.3413	0.3812	0.6587
5	0.4357	0.3293	0.4870	0.4192	0.4212	0.3812	0.4950	0.3513	0.3713	0.4431	0.6587
6	0.4389	0.3214	0.4830	0.4232	0.4251	0.4012	0.4850	0.3772	0.3733	0.4551	0.6447
7	0.4429	0.2994	0.4551	0.4152	0.4451	0.4251	0.4890	0.3892	0.3852	0.4711	0.6547
8	0.4419	0.2894	0.4591	0.4212	0.4471	0.4232	0.4790	0.4092	0.3713	0.4691	0.6507
9	0.4423	0.2974	0.4411	0.4271	0.4451	0.4232	0.4770	0.4052	0.3812	0.4691	0.6567
10	0.4405	0.2914	0.4132	0.4232	0.4371	0.4192	0.4830	0.4132	0.3852	0.4830	0.6567
11	0.4407	0.2914	0.4112	0.4172	0.4331	0.4132	0.4770	0.4291	0.3892	0.4910	0.6547
12	0.4371	0.3094	0.4012	0.4072	0.4172	0.4152	0.4691	0.4251	0.3733	0.4890	0.6647
13	0.4357	0.3234	0.3912	0.4012	0.4192	0.4092	0.4770	0.4132	0.3633	0.4950	0.6647
14	0.4353	0.3433	0.3752	0.3972	0.4032	0.4152	0.4691	0.4152	0.3633	0.5010	0.6707
15	0.4389	0.3393	0.3832	0.4012	0.4052	0.4152	0.4770	0.4251	0.3713	0.5030	0.6687
16	0.4391	0.3293	0.3693	0.3912	0.4072	0.4112	0.4870	0.4192	0.3772	0.5130	0.6866
17	0.4291	0.3194	0.3453	0.3912	0.3852	0.4032	0.4810	0.4132	0.3673	0.5090	0.6766
18	0.4293	0.3273	0.3453	0.3912	0.3772	0.3972	0.4571	0.4291	0.3673	0.5110	0.6906
19	0.4287	0.3373	0.3293	0.3812	0.3872	0.4012	0.4571	0.4232	0.3633	0.5130	0.6946
20	0.4250	0.3333	0.3214	0.3772	0.3832	0.3952	0.4551	0.4271	0.3533	0.5070	0.6966
21	0.4166	0.3293	0.3034	0.3693	0.3752	0.3892	0.4491	0.4152	0.3453	0.5110	0.6786
22	0.4156	0.3433	0.2854	0.3653	0.3772	0.3852	0.4431	0.4172	0.3513	0.5130	0.6747
23	0.4116	0.3453	0.2774	0.3633	0.3693	0.3792	0.4471	0.4072	0.3393	0.5130	0.6747
24	0.4114	0.3453	0.2814	0.3673	0.3673	0.3792	0.4371	0.3972	0.3433	0.5130	0.6826
25	0.4062	0.3333	0.2854	0.3653	0.3553	0.3733	0.4172	0.3952	0.3373	0.5150	0.6846
26	0.4040	0.3473	0.2994	0.3573	0.3573	0.3713	0.4032	0.3872	0.3313	0.5090	0.6766
27	0.3984	0.3413	0.2994	0.3433	0.3413	0.3613	0.4052	0.3812	0.3353	0.5110	0.6647
28	0.4018	0.3573	0.3214	0.3433	0.3393	0.3573	0.3972	0.3892	0.3313	0.5170	0.6647
29	0.3986	0.3513	0.3253	0.3373	0.3234	0.3513	0.3952	0.3832	0.3273	0.5170	0.6747
30	0.4002	0.3613	0.3433	0.3154	0.3114	0.3593	0.4052	0.3852	0.3313	0.5170	0.6727
31	0.3998	0.3553	0.3673	0.3134	0.3114	0.3533	0.4012	0.3772	0.3234	0.5210	0.6747
32	0.3960	0.3533	0.3792	0.3094	0.3054	0.3493	0.3812	0.3673	0.3253	0.5190	0.6707
33	0.3942	0.3573	0.4032	0.3074	0.2934	0.3433	0.3733	0.3533	0.3253	0.5230	0.6627
34	0.3944	0.3693	0.4192	0.3074	0.2894	0.3433	0.3633	0.3513	0.3174	0.5230	0.6607
35	0.3920	0.3613	0.4371	0.3034	0.2794	0.3433	0.3593	0.3513	0.3094	0.5230	0.6527
36	0.3910	0.3533	0.4571	0.3054	0.2735	0.3413	0.3473	0.3533	0.3074	0.5230	0.6487
37	0.3928	0.3433	0.4850	0.2974	0.2854	0.3353	0.3473	0.3533	0.3054	0.5250	0.6507
38	0.3920	0.3453	0.4990	0.2914	0.2854	0.3313	0.3473	0.3553	0.2994	0.5230	0.6427
39	0.3812	0.3433	0.5130	0.2874	0.2695	0.3293	0.3393	0.3473	0.2934	0.5250	0.5649
40	0.3794	0.3413	0.5269	0.2834	0.2695	0.3253	0.3293	0.3413	0.2994	0.5289	0.5489
41	0.3782	0.3433	0.5469	0.2794	0.2635	0.3234	0.3134	0.3373	0.3054	0.5289	0.5409
42	0.3796	0.3433	0.5669	0.2814	0.2575	0.3214	0.3154	0.3393	0.3034	0.5289	0.5389
43	0.3788	0.3453	0.5908	0.2774	0.2555	0.3134	0.3074	0.3253	0.3094	0.5289	0.5349
44	0.3776	0.3513	0.6168	0.2754	0.2515	0.3034	0.3054	0.3234	0.2874	0.5210	0.5409

kNN, 10 labels, PCA dimensionality reduction

	Overall Accuracy	#16andpregnant	#cybermonday	#england2018	#goblue	#imaceleb	#losemymind	#thingsthatpissmeoff	#welldone	#wenottogether	#why90srocked
45	0.3780	0.3433	0.6248	0.2635	0.2495	0.2954	0.3034	0.3533	0.2794	0.5250	0.5429
46	0.3778	0.3473	0.6407	0.2595	0.2455	0.2954	0.2994	0.3493	0.2735	0.5269	0.5409
47	0.3762	0.3373	0.6547	0.2535	0.2455	0.2874	0.2954	0.3473	0.2715	0.5289	0.5409
48	0.3782	0.3293	0.6766	0.2555	0.2455	0.2834	0.2974	0.3513	0.2794	0.5269	0.5369
49	0.3792	0.3333	0.6966	0.2515	0.2555	0.2814	0.2894	0.3473	0.2774	0.5250	0.5349
50	0.3812	0.3353	0.7186	0.2395	0.2715	0.2735	0.2834	0.3533	0.2794	0.5250	0.5329
51	0.3784	0.3333	0.7325	0.2395	0.2774	0.2675	0.2635	0.3453	0.2774	0.5250	0.5230
52	0.3796	0.3313	0.7305	0.2395	0.2814	0.2655	0.2635	0.3553	0.2854	0.5250	0.5190
53	0.3826	0.3353	0.7385	0.2355	0.2934	0.2595	0.2655	0.3673	0.2934	0.5210	0.5170
54	0.3850	0.3393	0.7465	0.2315	0.3054	0.2615	0.2595	0.3832	0.2954	0.5190	0.5090
55	0.3864	0.3373	0.7525	0.2315	0.3194	0.2535	0.2615	0.3952	0.2914	0.5170	0.5050
56	0.3876	0.3273	0.7505	0.2295	0.3413	0.2555	0.2695	0.3992	0.2814	0.5170	0.5050
57	0.3876	0.3234	0.7345	0.2275	0.3433	0.2555	0.2695	0.4172	0.2834	0.5190	0.5030
58	0.3880	0.3273	0.7106	0.2255	0.3473	0.2535	0.2735	0.4331	0.2814	0.5190	0.5090
59	0.3898	0.3253	0.7046	0.2236	0.3573	0.2535	0.2794	0.4411	0.2854	0.5190	0.5090
60	0.3892	0.3234	0.6826	0.2216	0.3573	0.2515	0.2814	0.4571	0.2874	0.5170	0.5130
61	0.3876	0.3234	0.6587	0.2216	0.3533	0.2475	0.2954	0.4691	0.2854	0.5170	0.5050
62	0.3876	0.3154	0.6387	0.2216	0.3493	0.2435	0.3014	0.4930	0.2834	0.5210	0.5090
63	0.3858	0.3214	0.6068	0.2196	0.3553	0.2395	0.3074	0.5050	0.2834	0.5130	0.5070
64	0.3824	0.3154	0.5828	0.2156	0.3493	0.2355	0.3054	0.5210	0.2814	0.5170	0.5010
65	0.3812	0.3174	0.5629	0.2136	0.3353	0.2375	0.3214	0.5369	0.2754	0.5150	0.4970
66	0.3780	0.3094	0.5369	0.2116	0.3453	0.2355	0.3194	0.5369	0.2774	0.5130	0.4950
67	0.3758	0.3074	0.5070	0.2096	0.3533	0.2335	0.3194	0.5429	0.2794	0.5130	0.4930
68	0.3705	0.2954	0.4750	0.2096	0.3453	0.2335	0.3054	0.5509	0.2834	0.5130	0.4930
69	0.3677	0.2874	0.4411	0.2096	0.3513	0.2315	0.2974	0.5689	0.2834	0.5150	0.4910
70	0.3639	0.2774	0.4152	0.2116	0.3433	0.2315	0.2994	0.5808	0.2774	0.5130	0.4890
71	0.3627	0.2754	0.4092	0.2076	0.3433	0.2295	0.2914	0.5928	0.2735	0.5130	0.4910
72	0.3585	0.2655	0.3812	0.2056	0.3333	0.2275	0.2914	0.6008	0.2754	0.5150	0.4890
73	0.3543	0.2555	0.3633	0.2016	0.3214	0.2255	0.2854	0.6208	0.2675	0.5170	0.4850
74	0.3513	0.2515	0.3393	0.1996	0.3074	0.2216	0.2874	0.6327	0.2655	0.5190	0.4890
75	0.3485	0.2455	0.3293	0.1976	0.2934	0.2216	0.2854	0.6427	0.2655	0.5190	0.4850
76	0.3429	0.2415	0.3014	0.1956	0.2854	0.2196	0.2774	0.6547	0.2595	0.5170	0.4770
77	0.3397	0.2375	0.2834	0.1936	0.2834	0.2156	0.2774	0.6627	0.2535	0.5170	0.4731
78	0.3353	0.2295	0.2595	0.1956	0.2754	0.2156	0.2635	0.6747	0.2495	0.5170	0.4731
79	0.3271	0.2196	0.2435	0.1896	0.2575	0.2016	0.2475	0.6846	0.2455	0.5170	0.4651
80	0.3224	0.2176	0.2275	0.1836	0.2355	0.1976	0.2415	0.7006	0.2455	0.5130	0.4611
81	0.3110	0.2116	0.2176	0.1737	0.2295	0.1976	0.2395	0.7106	0.2395	0.5110	0.3792
82	0.3072	0.2116	0.1956	0.1756	0.2216	0.1996	0.2116	0.7246	0.2395	0.5170	0.3752
83	0.2980	0.2036	0.1796	0.1717	0.2176	0.1976	0.2096	0.7345	0.2355	0.5190	0.3114
84	0.2932	0.2056	0.1537	0.1717	0.2036	0.1896	0.2076	0.7425	0.2335	0.5190	0.3054
85	0.2894	0.2016	0.1497	0.1677	0.1896	0.1876	0.2076	0.7505	0.2275	0.5150	0.2974
86	0.2870	0.2036	0.1397	0.1637	0.1836	0.1776	0.2036	0.7585	0.2255	0.5150	0.2994
87	0.2804	0.1956	0.1257	0.1557	0.1737	0.1796	0.1916	0.7605	0.2136	0.5070	0.3014
88	0.2772	0.1916	0.1218	0.1497	0.1657	0.1796	0.1856	0.7625	0.2116	0.5050	0.2994

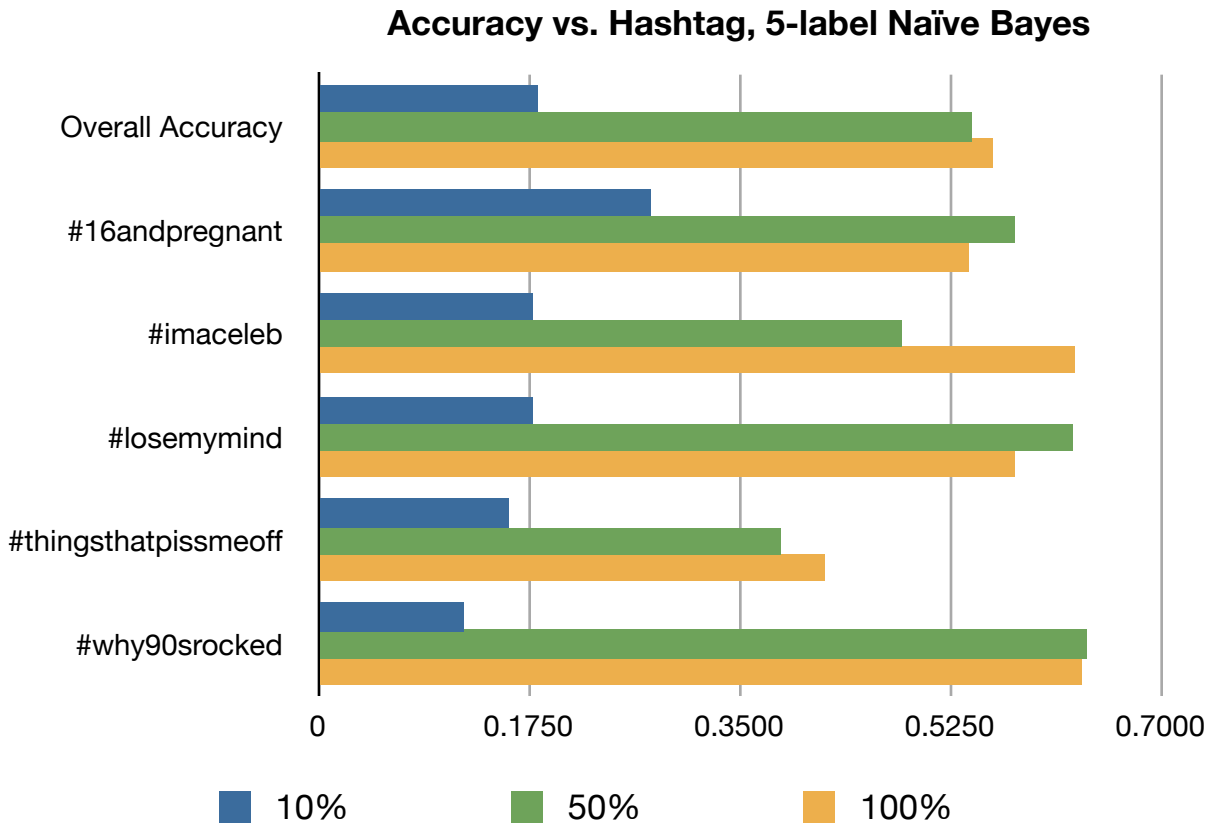
kNN, 10 labels, PCA dimensionality reduction

	Overall Accuracy	#16andpregnant	#cybermonday	#england2018	#goblue	#imaceleb	#losemymind	#thingsthatpissmeoff	#welldone	#wenottogether	#why90srocked
89	0.2749	0.1876	0.1078	0.1477	0.1577	0.1796	0.1876	0.7685	0.2156	0.5030	0.2934
90	0.2711	0.1796	0.0998	0.1417	0.1597	0.1737	0.1836	0.7745	0.2096	0.4990	0.2894
91	0.2659	0.1737	0.0938	0.1397	0.1517	0.1677	0.1737	0.7784	0.1976	0.4990	0.2834
92	0.2635	0.1737	0.0858	0.1357	0.1497	0.1617	0.1657	0.7904	0.1936	0.4990	0.2794
93	0.2611	0.1637	0.0818	0.1357	0.1397	0.1637	0.1637	0.7984	0.1936	0.4970	0.2735
94	0.2557	0.1537	0.0778	0.1337	0.1277	0.1597	0.1577	0.8024	0.1836	0.4930	0.2675
95	0.2531	0.1517	0.0719	0.1317	0.1198	0.1577	0.1537	0.8044	0.1816	0.4930	0.2655
96	0.2503	0.1477	0.0679	0.1297	0.1078	0.1597	0.1477	0.8164	0.1836	0.4810	0.2615
97	0.2461	0.1477	0.0639	0.1277	0.1038	0.1517	0.1437	0.8124	0.1816	0.4731	0.2555
98	0.2451	0.1417	0.0599	0.1257	0.1038	0.1477	0.1457	0.8224	0.1796	0.4671	0.2575
99	0.2451	0.1417	0.0599	0.1277	0.0978	0.1457	0.1457	0.8363	0.1796	0.4611	0.2555
100	0.2439	0.1417	0.0599	0.1238	0.0958	0.1437	0.1457	0.8423	0.1776	0.4571	0.2515



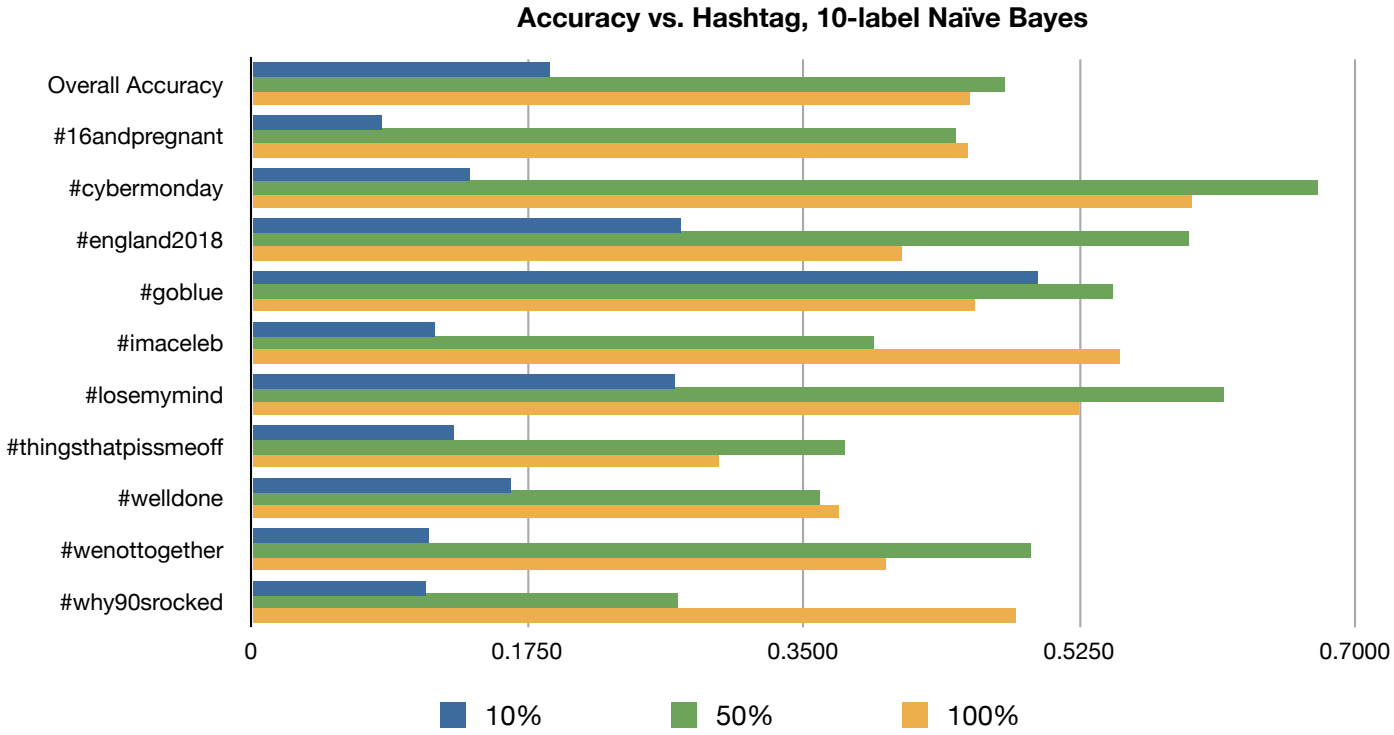
Naive Bayes, 5 labels

	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
10%	0.1820	0.2760	0.1780	0.1780	0.1580	0.1200
50%	0.5420	0.5780	0.4840	0.6260	0.3840	0.6380
100%	0.5600	0.5400	0.6280	0.5780	0.4200	0.6340

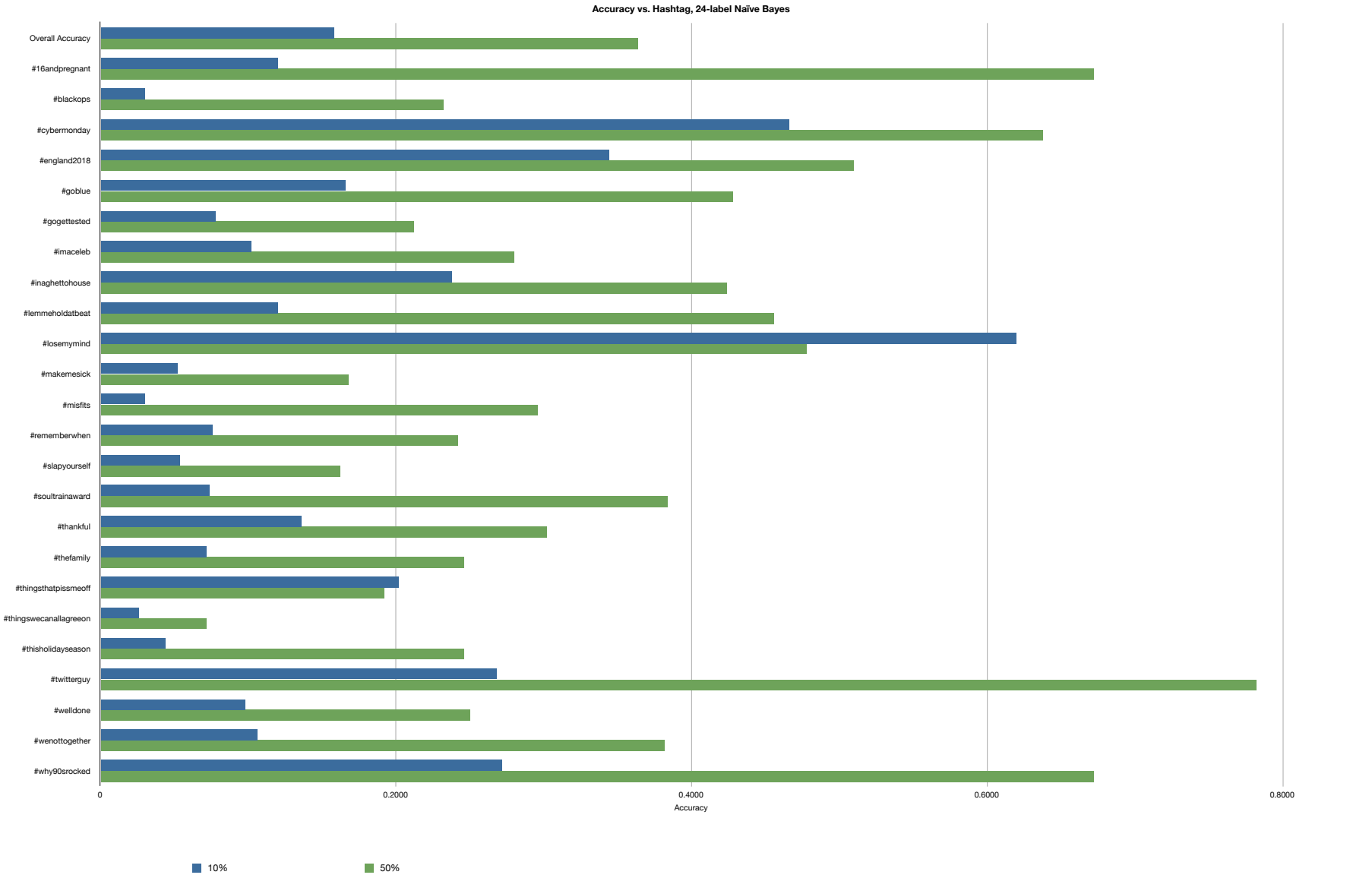


Naive Bayes, 10 labels

	Overall Accuracy	#16andpregnant	#cybermonday	#england2018	#goblue	#imaceleb	#losemymind	#thingsthatpissmeoff	#welldone	#wenottogether	#why90srocked
10%	0.1888	0.0820	0.1380	0.2720	0.4980	0.1160	0.2680	0.1280	0.1640	0.1120	0.1100
50%	0.4772	0.4460	0.6760	0.5940	0.5460	0.3940	0.6160	0.3760	0.3600	0.4940	0.2700
100%	0.4548	0.4540	0.5960	0.4120	0.4580	0.5500	0.5240	0.2960	0.3720	0.4020	0.4840

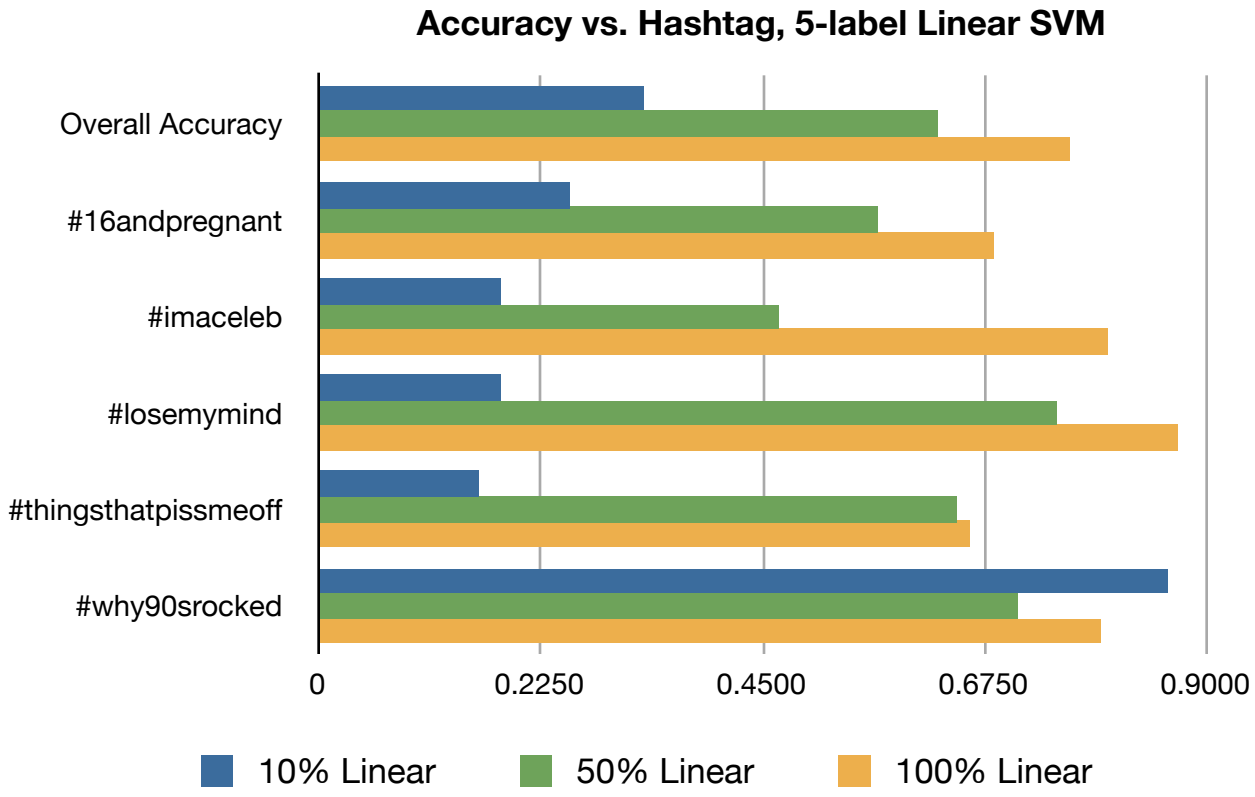


	Overall Accuracy	#16andpregnant	#blackops	#cybermonday	#england2018	#goblue	#gogettested	#imaceleb	#inaghettohouse	#lemmeholdatbeat	#losemymind	#makemesick	#misfits	#rememberwhen	#slapyourself	#soultrainaward	#thankful	#thefamily	#thingsthatpissmeoff	#thingswecanallagreeon	#thisholidayseason	#twitterguy	#welldone	#wenottogether	#why90srocked
10%	0.1581	0.1200	0.0300	0.4660	0.3440	0.1660	0.0780	0.1020	0.2380	0.1200	0.6200	0.0520	0.0300	0.0760	0.0540	0.0740	0.1360	0.0720	0.2020	0.0260	0.0440	0.2680	0.0980	0.1060	0.2720
50%	0.3636	0.6720	0.2320	0.6380	0.5100	0.4280	0.2120	0.2800	0.4240	0.4560	0.4780	0.1680	0.2960	0.2420	0.1620	0.3840	0.3020	0.2460	0.1920	0.0720	0.2460	0.7820	0.2500	0.3820	0.6720

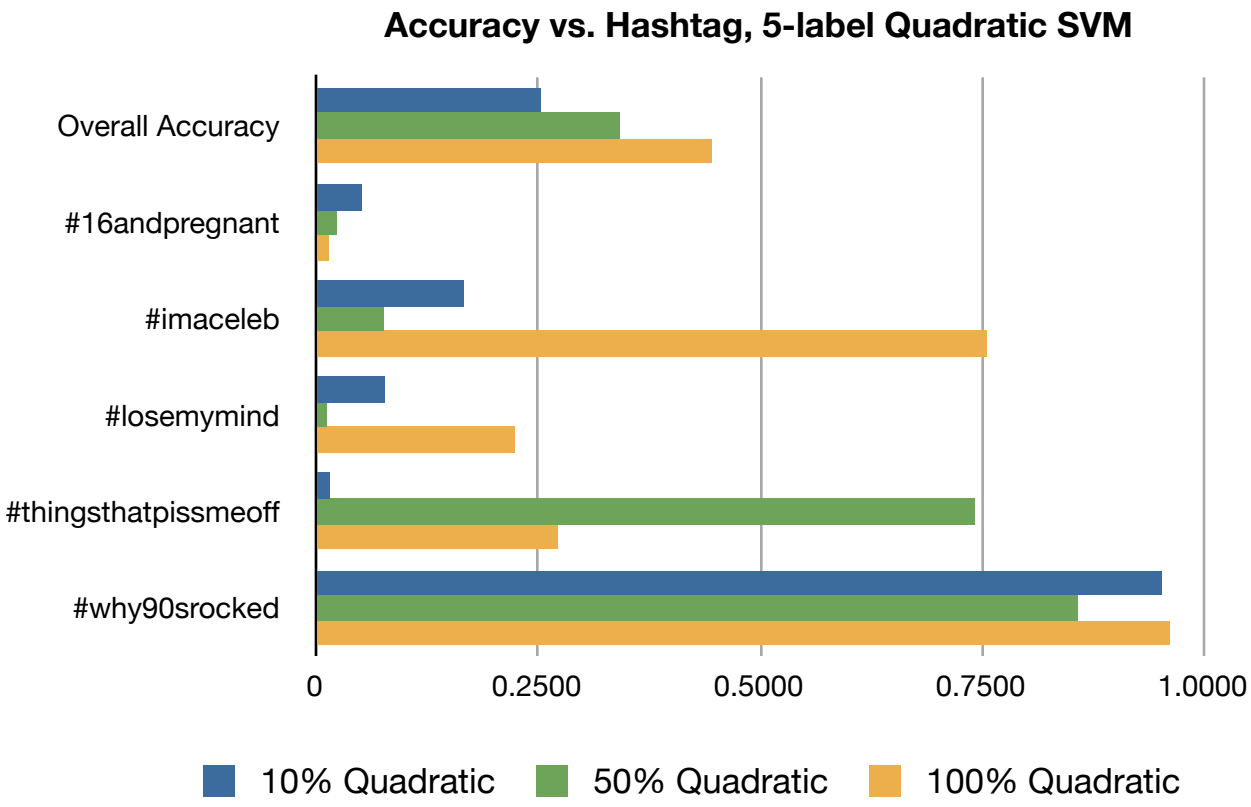


SVM, dictionary size dimensionality reduction, 5 labels

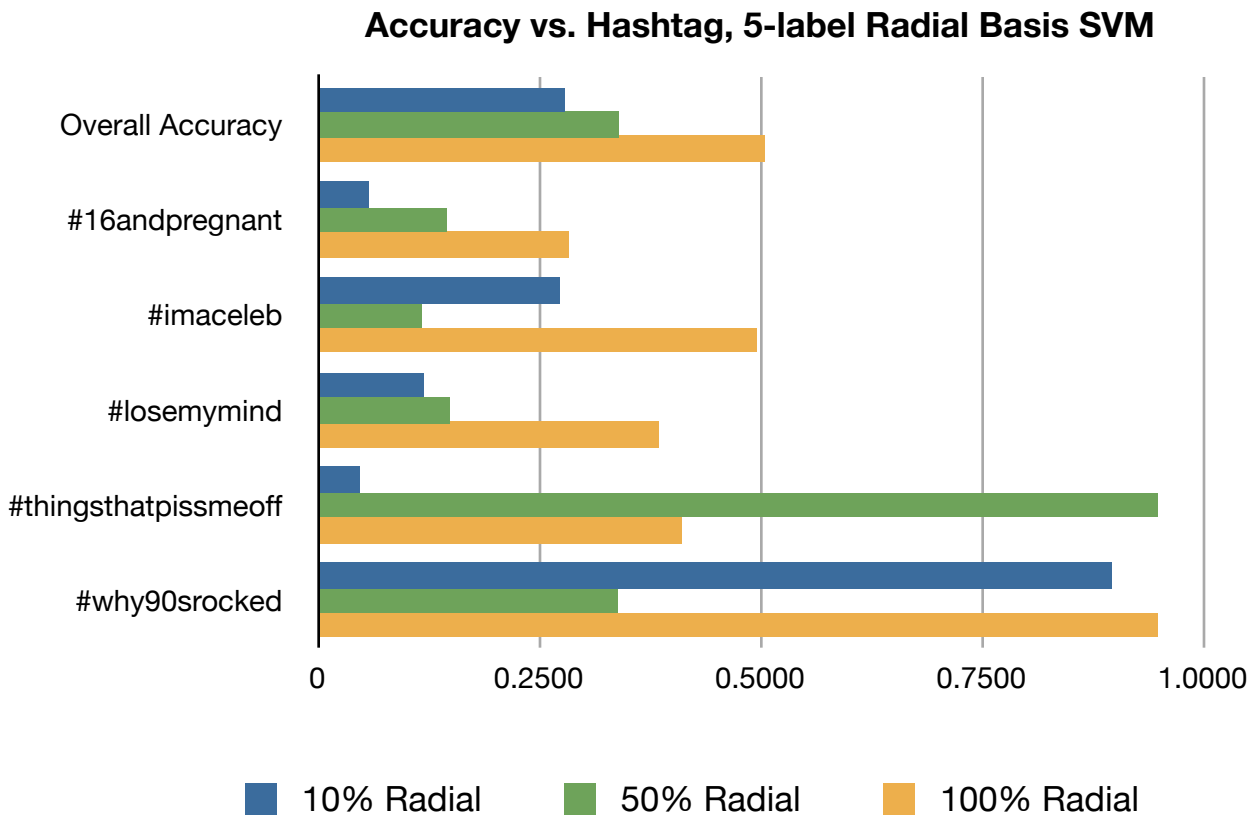
	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
10% Linear	0.3288	0.2540	0.1840	0.1840	0.1620	0.8600
10% Quadratic	0.2528	0.0520	0.1660	0.0780	0.0160	0.9520
10% Radial	0.2776	0.0560	0.2720	0.1180	0.0460	0.8960
50% Linear	0.6268	0.5660	0.4660	0.7480	0.6460	0.7080
50% Quadratic	0.3424	0.0240	0.0760	0.0120	0.7420	0.8580
50% Radial	0.3388	0.1440	0.1160	0.1480	0.9480	0.3380
100% Linear	0.7612	0.6840	0.8000	0.8700	0.6600	0.7920
100% Quadratic	0.4456	0.0140	0.7560	0.2240	0.2720	0.9620
100% Radial	0.5036	0.2820	0.4940	0.3840	0.4100	0.9480



SVM, dictionary size dimensionality reduction, 5 labels

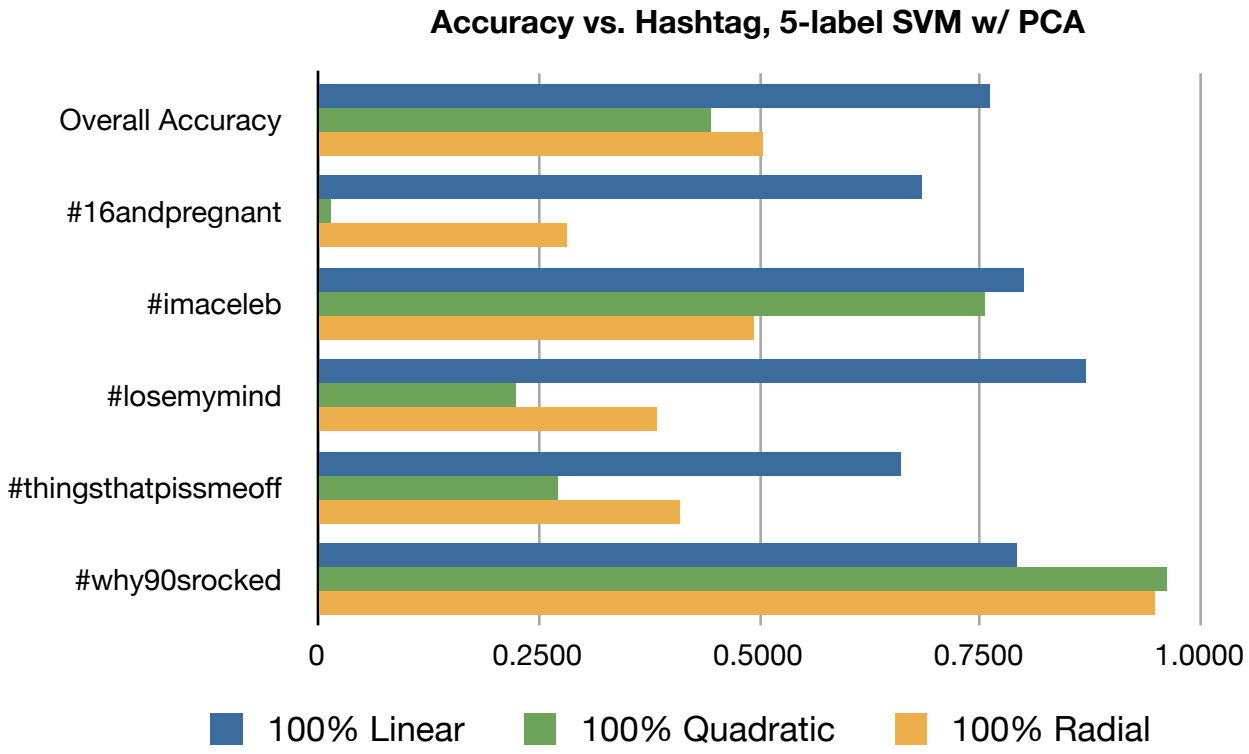


SVM, dictionary size dimensionality reduction, 5 labels



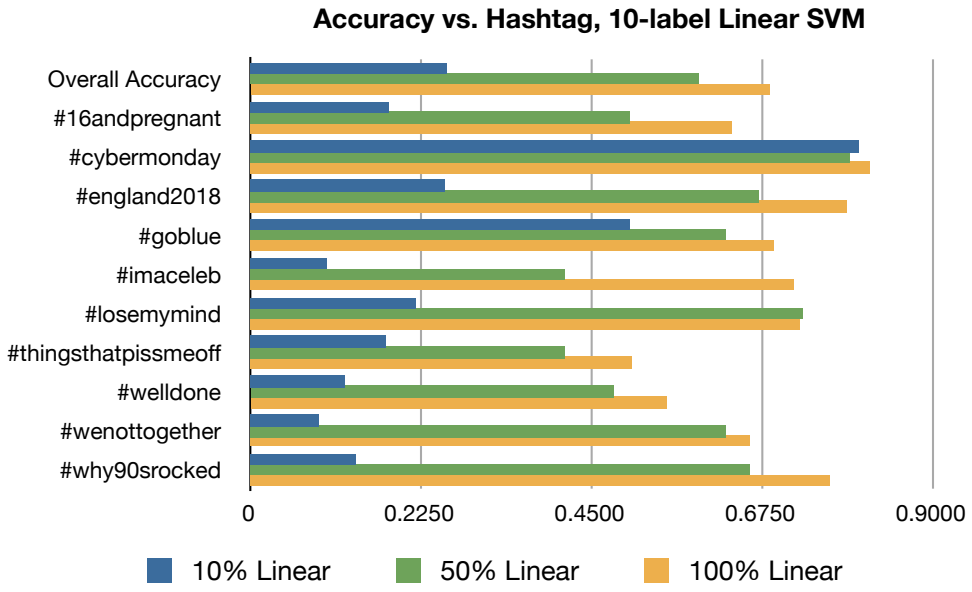
SVM, PCA dimensionality reduction, 5 labels

	Overall Accuracy	#16andpregnant	#imaceleb	#losemymind	#thingsthatpissmeoff	#why90srocked
100% Linear	0.7612	0.6840	0.8000	0.8700	0.6600	0.7920
100% Quadratic	0.4456	0.0140	0.7560	0.2240	0.2720	0.9620
100% Radial	0.5036	0.2820	0.4940	0.3840	0.4100	0.9480

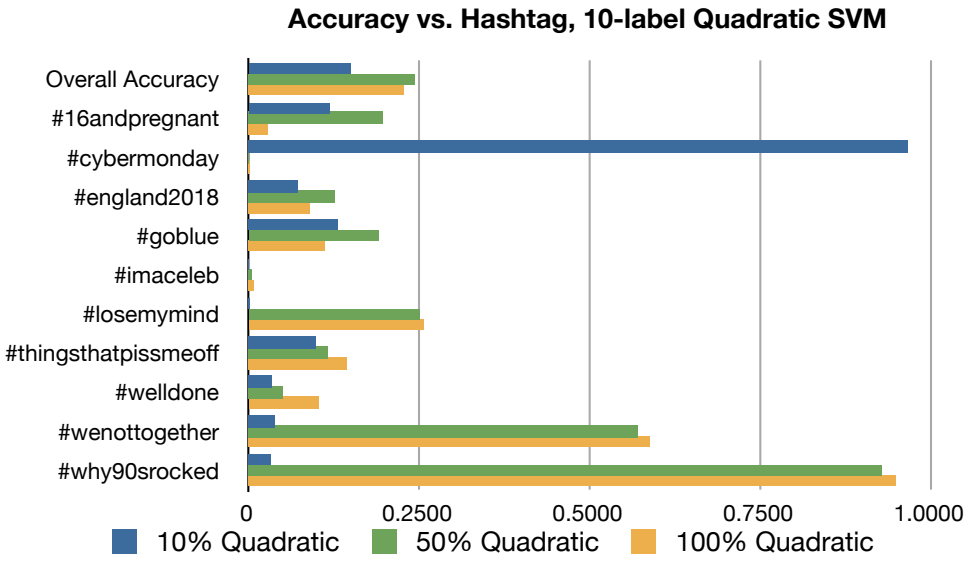


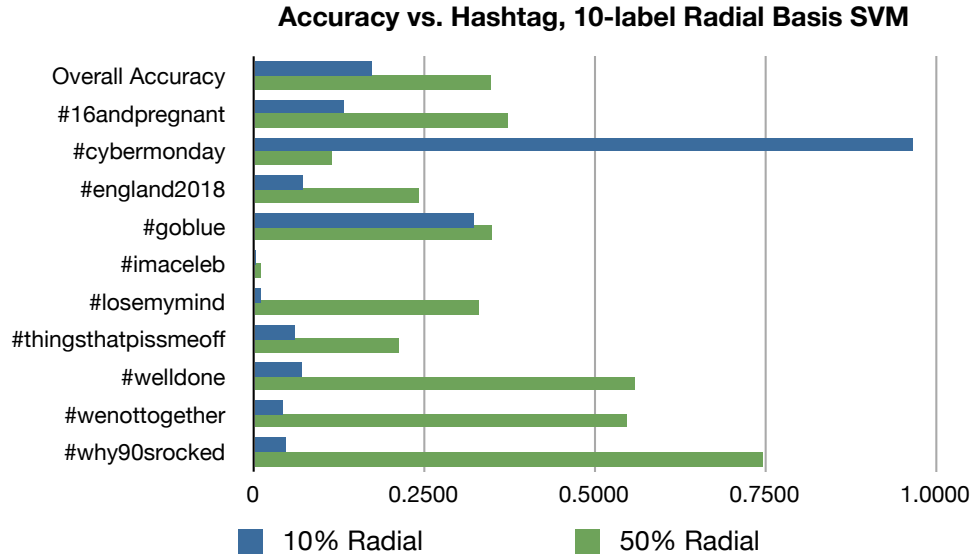
SVM, dictionary size dimensionality reduction, 10 labels

	Overall Accuracy	#16andpregnant	#cybermonday	#england2018	#goblue	#imaceleb	#losemymind	#thingsthatpissmeoff	#welldone	#wenottogether	#why90srocked
10% Linear	0.2608	0.1840	0.8040	0.2580	0.5020	0.1020	0.2200	0.1800	0.1260	0.0920	0.1400
10% Quadratic	0.1510	0.1200	0.9680	0.0740	0.1320	0.0020	0.0040	0.1000	0.0360	0.0400	0.0340
10% Radial	0.1722	0.1320	0.9660	0.0720	0.3220	0.0020	0.0100	0.0600	0.0700	0.0420	0.0460
50% Linear	0.5924	0.5020	0.7920	0.6720	0.6280	0.4160	0.7300	0.4160	0.4800	0.6280	0.6600
50% Quadratic	0.2450	0.1980	0.0020	0.1280	0.1920	0.0060	0.2520	0.1180	0.0520	0.5720	0.9300
50% Radial	0.3478	0.3720	0.1140	0.2420	0.3480	0.0100	0.3300	0.2120	0.5580	0.5460	0.7460
100% Linear	0.6858	0.6360	0.8180	0.7880	0.6920	0.7180	0.7260	0.5040	0.5500	0.6600	0.7660
100% Quadratic	0.2298	0.0300	0.0040	0.0920	0.1140	0.0100	0.2580	0.1460	0.1040	0.5900	0.9500



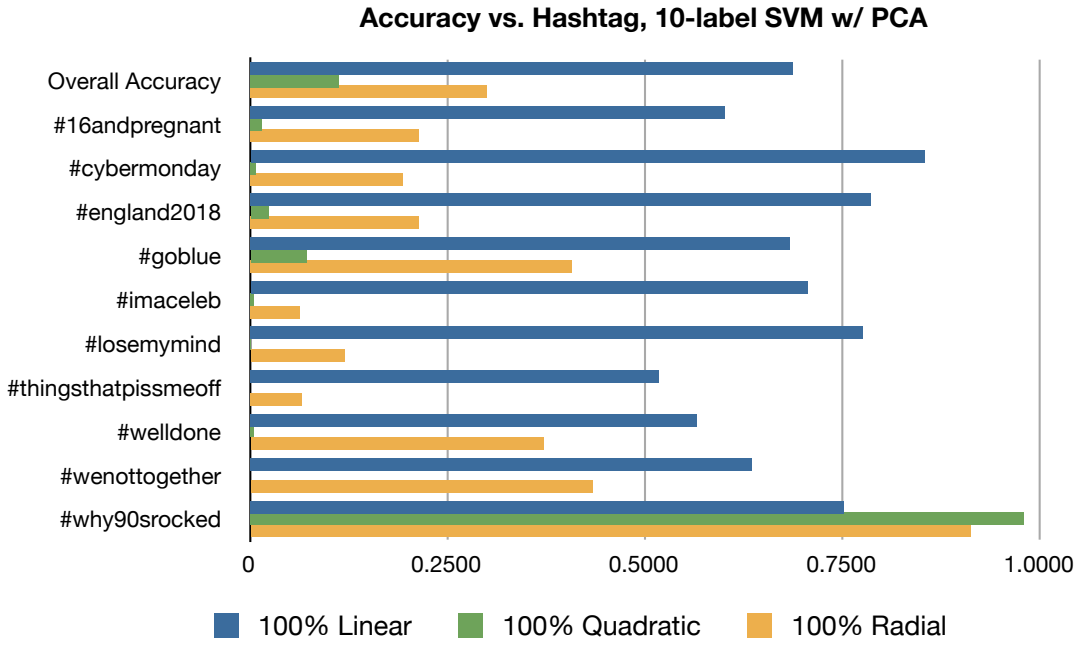
SVM, dictionary size dimensionality reduction, 10 labels





SVM, PCA dimensionality reduction, 10 labels

	Overall Accuracy	#16andpregnant	#cybermonday	#england2018	#goblue	#imaceleb	#losemymind	#thingsthatpissmeoff	#welldone	#wenottogether	#why90srocked
100% Linear	0.6880	0.6020	0.8540	0.7860	0.6840	0.7060	0.7760	0.5180	0.5660	0.6360	0.7520
100% Quadratic	0.1132	0.0160	0.0080	0.0240	0.0720	0.0060	0.0020	0.0000	0.0060	0.0000	0.9800
100% Radial	0.3008	0.2140	0.1940	0.2140	0.4080	0.0640	0.1200	0.0660	0.3720	0.4340	0.9120



AUTOMATIC TWEET HASHTAG CATEGORIZATION

Dan Schultz
Sunny Jolly

Introduction

slifty Dan Schultz
I'm working on a project for class that will automatically suggest tags for tweets. Lets see how hard I fail... #patternrecognition
3 hours ago

- Twitter is a micro-blogging platform (tweets limited to 140 characters)
- Users self-categorize tweets using hashtags
- We intend to provide a proof-of-concept capability for automatic hashtag categorization

SanityBuff NewsPoliticsInfo
Body scanner makers doubled #lobbying #money over 5 years - http://usat.ly/dd8lVu - **#TSA** #politics
7 minutes ago

TexasChelsey Chelsey
tomorrow at the airport I will face a terrible decision... pat down or scanner?
4 hours ago

flyingwithfish Flying With Fish by AlNilsen
Now On Flying With Fish - **#TSA** Clarifies Policy For Pat Down Refusal In Time For Thanksgiving - http://bit.ly/g9uzqU #airline #travel #lp
34 minutes ago

Motivation

- Hashtags often reveal subtle undertones normally inaccessible to computers
  - This opens up potential opportunities for natural language processing

- Promote and enforce consistency amongst popular tags
  - e.g., #LondonFire vs. #FireOfLondon vs. #LDNFire
- Promotes hashtag-based social networking

Dataset Generation

- Twitter API allows for pulling trending tags and 1500 most recent tweets per tag
- Filter out URLs and usernames from collected tweets and ensure no duplicate or foreign-language tweets
- Bag-of-words feature model (i.e., examining terminology rather than semantic meaning)

#imaceleb	3060	
#royalwedding	3078	2010-11-16 11:03:16
#thingswomendontdoanymore	3101	2010-11-16 16:17:53
#confessiontime	3381	
#xfactor	3698	
#iamspartacus	4174	
#sheiswifey	4245	
#greatthingaboutfriends	5600	2010-11-15 16:31:39
#holdmyhand	6887	2010-11-15 16:31:39
#blackops	7810	
#nevertrust	10516	2010-11-15 16:31:39

Methods

Assess comparative performance of different tweet classification strategies:

- Rule-Based Classifier
  - Manual sets of rules concerning key words and phrases
- Naïve Bayes Classifier
  - Words are tokens, tokens are probabilistically related to tags:

$$p(T|H) = \prod_{i=1}^{N} p(w_i|H) \qquad \text{argmax}_H \, p(H)\, p(T|H)$$

- *k*-Nearest Neighbor Classifier
  - Find *k* most similar tweets in training set over entire feature space, majority vote amongst most similar tweets
- Decision Trees, Support Vector Machines, Maximum Entropy Modeling, Artificial Neural Networks?

Status and Concerns

- Current collected data:
  - 50 tags, 3,000-10,000 tweets per tag
  - dictionary of 20,000+ words across the dataset
- Current status:
  - bag-of-words feature model implemented
  - partial implementations of Naïve Bayes and *k*-Nearest Neighbor classifiers
- Concerns:
  - bag-of-words model provides no inherent notion of semantic meaning
  - misspelled and shortened words common in tweets

AUTOMATIC TWEET HASHTAG CATEGORIZATION

Dan Schultz
Sunny Jolly

Introduction

slifty Dan Schultz
I'm working on a project for class that will automatically suggest tags for tweets. Lets see how hard I fail... #patternrecognition
3 hours ago

- Twitter is a micro-blogging platform (tweets limited to 140 characters)
- Users self-categorize tweets using hashtags
- We intend to provide a proof-of-concept capability for automatic hashtag categorization

SanityBuff NewsPoliticsInfo
Body scanner makers doubled #lobbying #money over 5 years - http://usat.ly/dd8lVu - **#TSA** #politics
7 minutes ago

TexasChelsey Chelsey
tomorrow at the airport I will face a terrible decision... pat down or scanner?
4 hours ago

flyingwithfish Flying With Fish by AlNilsen
Now On Flying With Fish - **#TSA** Clarifies Policy For Pat Down Refusal In Time For Thanksgiving - http://bit.ly/g9uzqU #airline #travel #lp
34 minutes ago

Dataset Generation

- Previous data had to be scrapped because it wasn't clean.
- New algorithms removed all foreign and duplicate tweets, while also cleaning individual tweets to remove "RT", usernames, and urls.
- Pulled tweets from the Twitter API over two weeks.
- Final set contains 24 tags which each have 2000 or more tweets.
- 1500 training points, 500 test points, and 100 validation points per tag.

Final tag pool:

#goblue, #makemesick, #soultrainawards, #thingswecanallagreeon, #thingsthatpissmeoff, #england2018, #lemmieholdatbeat, #welldone, #losemymind, #16andpregnant, #cybermondy, #imaceleb, #misfits, #inaghettohouse, #twitterguy, #rememberwhen, #blackops, #thisholidayseason, #gogettested, #thefamily, #thankful, #slapyourself, #why90srocked, #wenottogether

Feature Selection

- Bag of Words -- that's a big feature space! (Full data set contained 26722 unique words)

Feature optimization Techniques:

- Removed the 30 most common words.
- Explored using 10%, 50%, and 100% of full dictionary.
- When cutting words, used the standard deviation of uses across categories as a metric (words used more equally across all categories were cut sooner).

Because some of our algorithms required tokens (binary features), we chose to avoid manipulative dimensionality reduction such as PCA.

Methods Overview

Assess comparative performance of different tweet classification strategies:

- Naïve Bayes Classifier
- *k*-Nearest Neighbor Classifier
- Support Vector Machines

For each method we attempted to explore effectiveness across

Category Counts: 5 categories, 10 categories, all (24) categories.

Dictionary Sizes: 10%, 50%, and 100% of the full dictionary size.

Naïve Bayes

Words are tokens, tokens are probabilistically related to tags

1. Training --> form a look-up table:

$$p(w_i|H) = \frac{\#\ of\ occurrences\ of\ w_i\ in\ hashtag\ H}{\#\ of\ occurrences\ of\ w_i\ in\ complete\ training\ set}$$

2. Calculate likelihoods:

$$p(T|H) = \prod_{i=1}^{N} p(w_i|H)$$

3. Calculate posterior probabilities and classify:

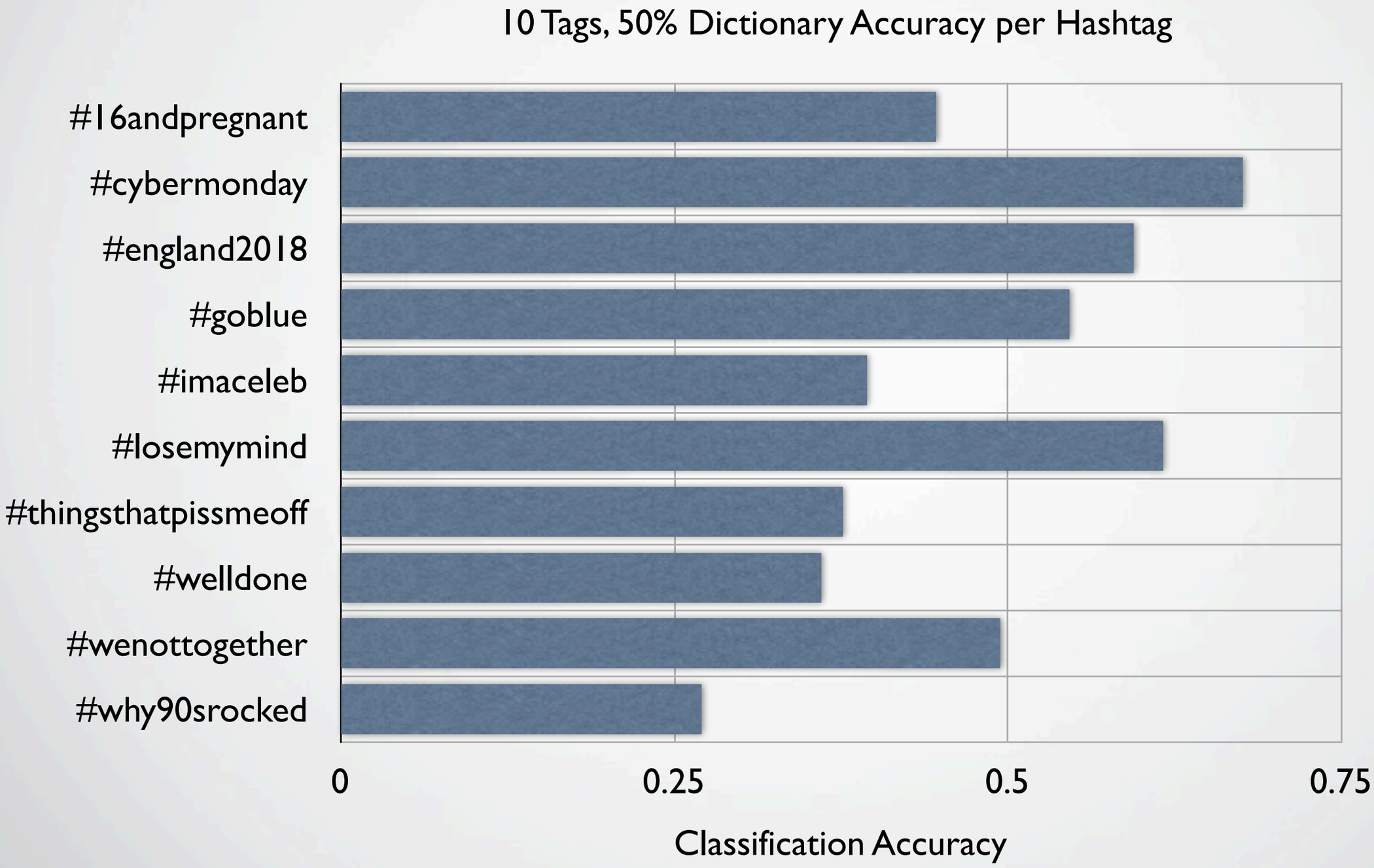
$$\text{argmax}_H\, p(H)\, p(T|H)$$

Naïve Bayes

Best Overall Accuracy %

	5 Tags	10 Tags	24 Tags
10% Dictionary	18%	19%	16%
50% Dictionary	54%	48%	--
100% Dictionary	56%	46%	--

Naïve Bayes


10 Tags, 50% Dictionary Accuracy per Hashtag
#16andpregnant
#cybermonday
#england2018
#goblue
#imaceleb
#losemymind
#thingsthatpissmeoff
#welldone
#wenottogether
#why90srocked
0
0.25
0.5
0.75
Classification Accuracy

k-Nearest Neighbor

Notes:

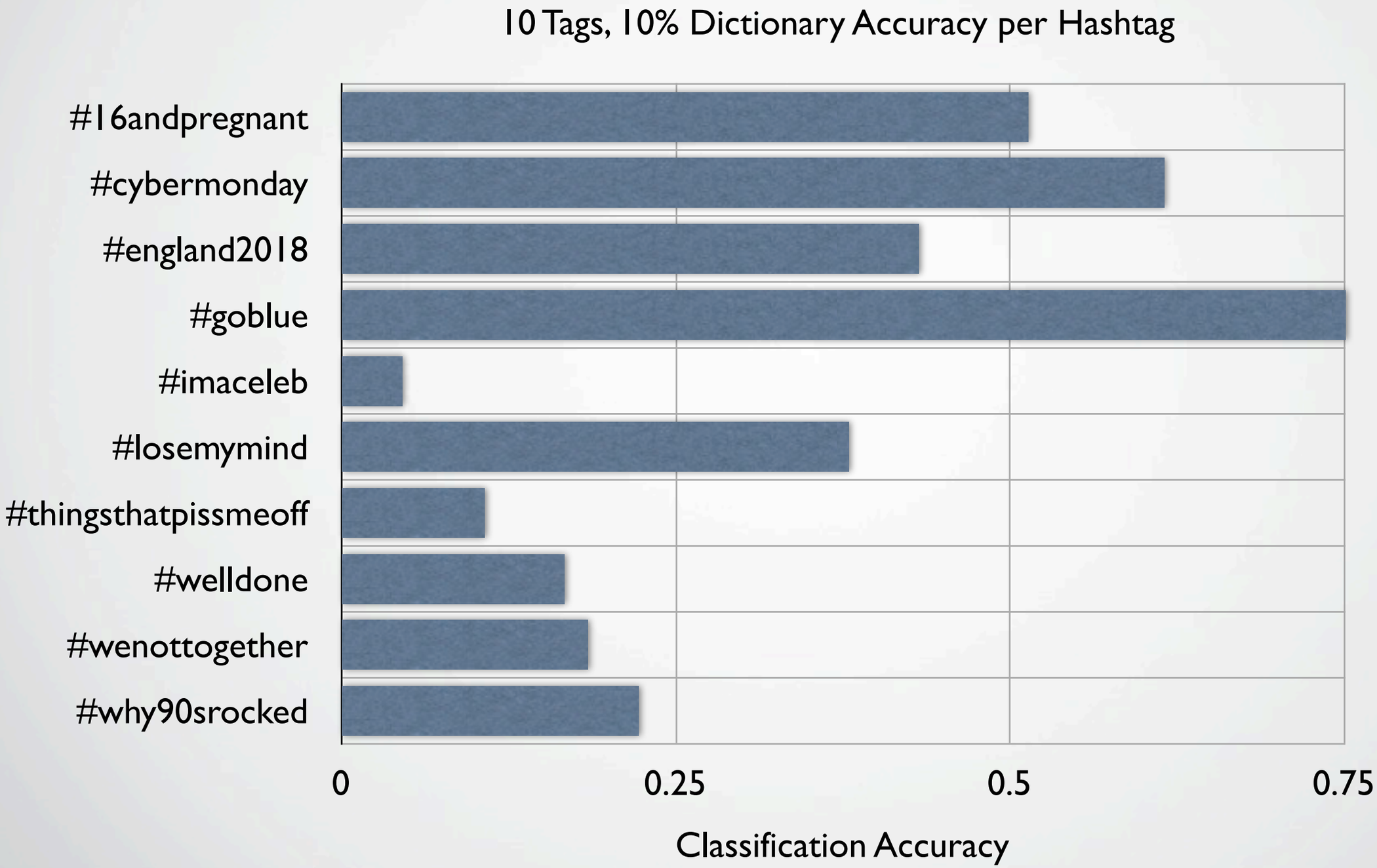
- Explored for K = 1...100
- We included an “ignore” metric: Neighbors that don’t share at least two words are not considered neighbors.
- If a test point doesn’t have enough neighbors, it is not categorized.
- As K increased, the number of attempted categorizations decreased while accuracy tended to increase.
- As dictionary sized increased, the number of attempted categorizations increased significantly.

k-Nearest Neighbor

Best Overall Accuracy % [Without Ignored (With Ignored)]

K=1	5 Tags	10 Tags	24 Tags
10% Dictionary	67% (2%)	47% (4%)	--
50% Dictionary	51% (35%)	--	--
100% Dictionary	61% (54%)	--	--

k-Nearest Neighbor



Support Vector Machines

Notes:

- Used LIBSVM library (courtesy Chih-Chung Chang and Chih-Jen Lin, National Taiwan University)
- Explored linear, quadratic (polynomial of degree 2), radial basis function kernels
  - Linear kernel: $K(x_i,x_j)=x_i^T x_j$
  - Quadratic kernel (γ = 1 / number features):

$$K(x_i,x_j)=(\gamma x_i^T x_j)^2$$

  - Radial basis function kernel (γ = 1 / number features):

$$K(x_i,x_j)=exp(\gamma||x_i - x_j||^2)$$

- No scaling of feature vectors needed -- feature values are already binary

Support Vector Machines

Best Overall Accuracy % (Linear Kernel)

	5 Tags	10 Tags	24 Tags
10% Dictionary	32%	26%	18%
50% Dictionary	63%	59%	--
100% Dictionary	76%	69%	--

Support Vector Machines

Best Overall Accuracy % (Quadratic Kernel)

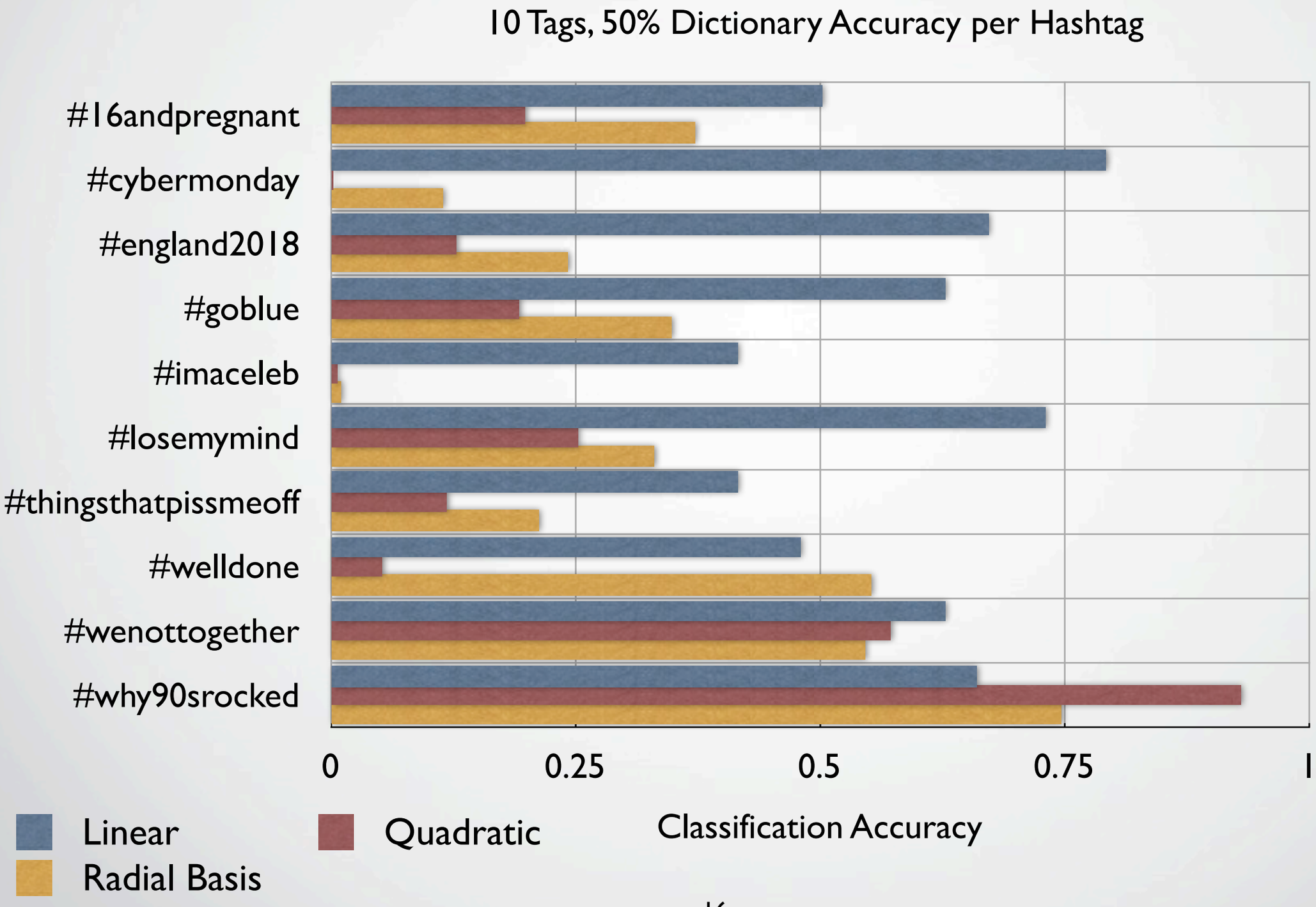
	5 Tags	10 Tags	24 Tags
10% Dictionary	25%	15%	--
50% Dictionary	34%	25%	--
100% Dictionary	44%	24%	--

Support Vector Machines

Best Overall Accuracy % (Radial Basis Function Kernel)

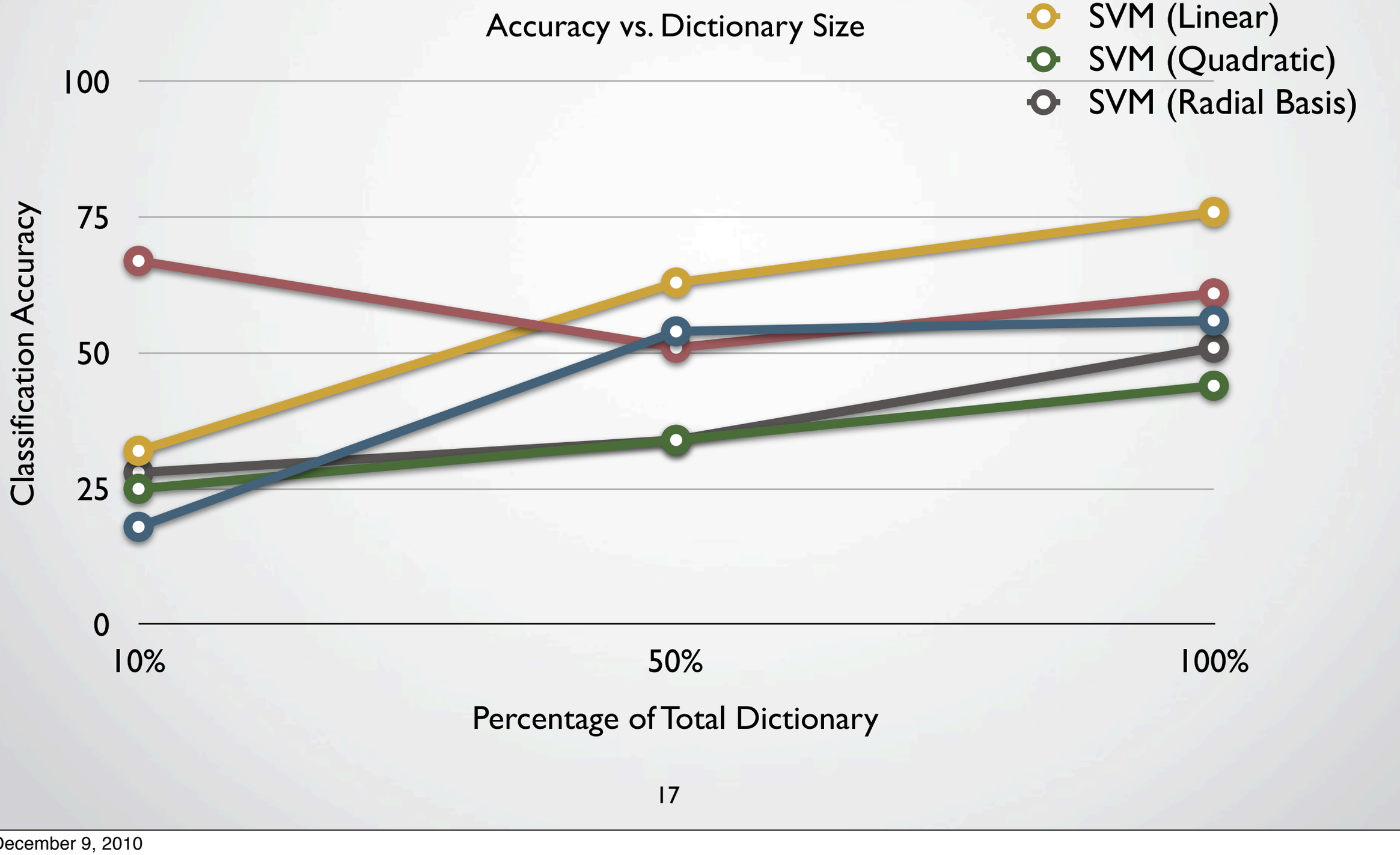
	5 Tags	10 Tags	24 Tags
10% Dictionary	28%	17%	--
50% Dictionary	34%	35%	--
100% Dictionary	51%	--	--

Support Vector Machines



Comparative Performance

Best Overall Accuracy % [5 tags]



Comparative Performance

Tags/ Dictionary	5/10%	5/50%	5/100%	10/10%	10/50%	10/100%	24/10%	24/50%	24/100%
Naïve Bayes	18%	54%	56%	19%	48%	46%	16%	--	--
kNN	67%	51%	61%	47%	--	--	--	--	--
SVM (Linear)	32%	63%	76%	26%	59%	69%	18%	--	--
SVM (Quadratic)	25%	34%	44%	15%	25%	24%	--	--	--
SVM (Radial)	28%	24%	51%	17%	35%	--	--	--	--

Suggestions for Future Work

- More data for training and testing needed to more adequately test algorithms
- More computing power / optimized algorithms needed to test larger dictionary sizes
- More “cleaning-up” of data to remove misspellings
- Bag-of-words should take into account relative importance of words (continuum of weights rather than binary values)
- More classification algorithms (e.g., decision trees, artificial neural networks)

Conclusions

- Best classification method: SVM with linear kernel
  - SVMs perform well in the text categorization problem due to their ability to learn for high-dimensional (e.g., 10000+) feature spaces and their ability to handle sparse instances[1]
- Lack of computing power was a major bottleneck
  - Design algorithms with parallelization capabilities in mind!
- More categories → less classification accuracy
- Greater dictionary size → greater classification accuracy

1. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the Tenth European Conference on Machine Learning (ECML '98), Lecture Notes in Computer Science*, Number 1398 (pp. 137–142).