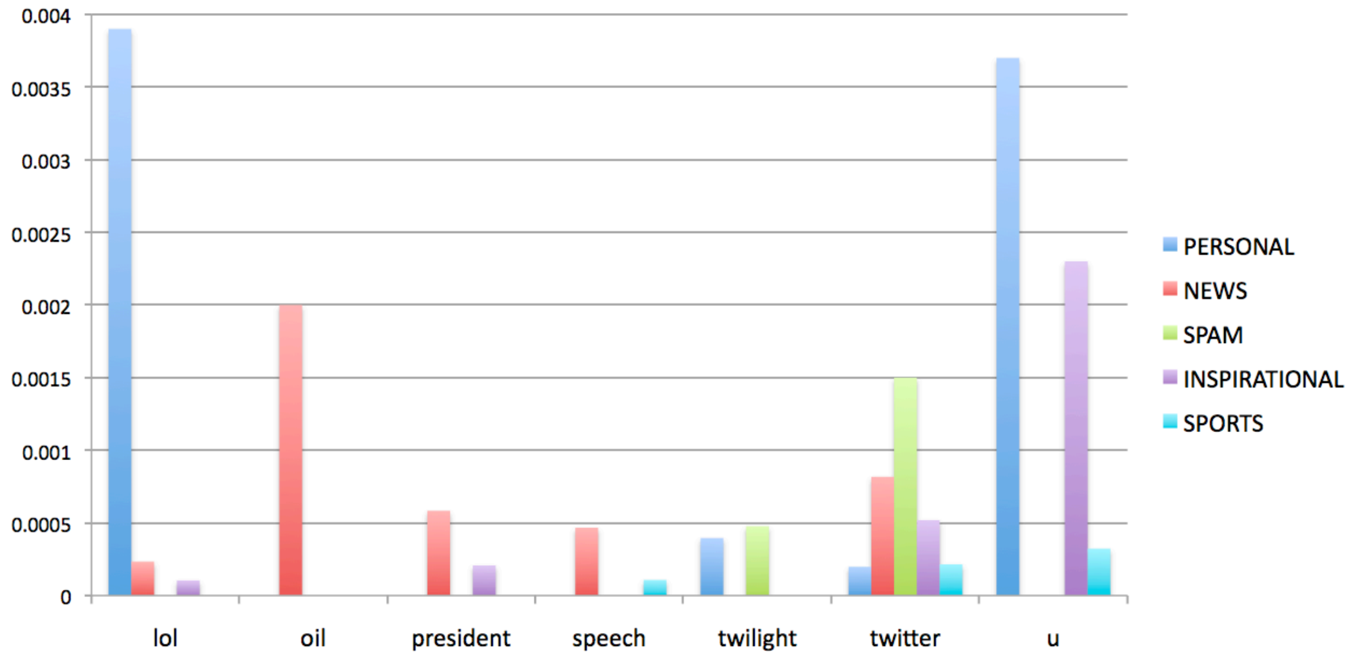
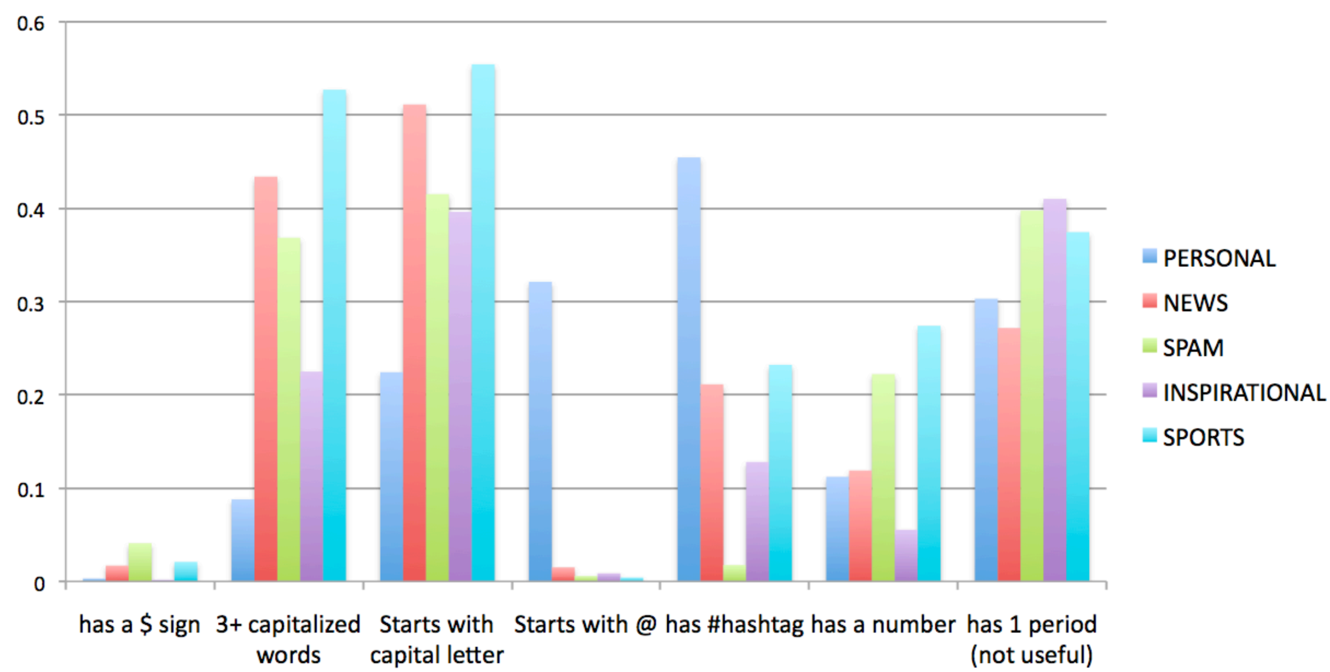
5.2 User Classification Scores

User	Classifier	Size	
twitter.com/cnn	NaiveBayes	2497	SmartCombined
	PERSONAL	0	
	NEWS	0.6428571	
	SPAM	0.14285714	
	INSPIRATIONAL	0	
	Sports	0.21428572	
	Score:	0.785714286	0.785714286
twitter.com/ihatequotes	NaiveBayes	2497	
	PERSONAL	0.2	
	NEWS	0	
	SPAM	0.13333333	
	INSPIRATIONAL	0.53333332	
	Sports	0.13333333	
	Score:	0.533333333	0.733333333
twitter.com/nba	NaiveBayes	2497	
	PERSONAL	0.07142857	
	NEWS	0.07142857	
	SPAM	0	
	INSPIRATIONAL	0	
	Sports	0.8571429	
	Score:	0.857142857	0.857142857
twitter.com/justinbieber	NaiveBayes	2497	
	PERSONAL	0.9285714	
	NEWS	0	
	SPAM	0.07142857	
	INSPIRATIONAL	0	
	Sports	0	
	Score:	0.928571429	0.857142857

5.3 Charts


0.004
0.0035
0.003
0.0025
0.002
0.0015
0.001
0.0005
0
lol
oil
president
speech
twilight
twitter
u
PERSONAL
NEWS
SPAM
INSPIRATIONAL
SPORTS


0.6
0.5
0.4
0.3
0.2
0.1
0
has a $ sign
3+ capitalized words
Starts with capital letter
Starts with @
has #hashtag
has a number
has 1 period (not useful)
PERSONAL
NEWS
SPAM
INSPIRATIONAL
SPORTS

6. Conclusions

6.1 Conclusions About Twitter From our Analysis

Much media hype surrounding Twitter has been with regard to the potential for real-time search; Twitter microbloggers can provide a number of instantaneous perspectives on any event around the world. This also means that new trends and topics are constantly appearing in any Twitter stream. Particularly for some of the categories chosen for this project (i.e. news, sports) classifiers will need to be constantly tuned and retrained to take such new vocab, hashtags and trends into account.

Furthermore, a sense of global context seems to permeate Twitter posts. A headline such as "Wow. That ump should be killed. For that mustache. (And also for the worst call ever.)" refers to a baseball game in which a highly controversial call had just been made. The lay reader would have great difficulty understanding this sports post, especially without a hashtag or incorporated link; but these are the headline-like restrictions imposed by Twitter's 140 character limitation. A much larger project would have to keep track of ongoing games and popular tags in order to best recognize and categorize such tweets.

It appears that Naïve Bayes performed the best of the simple classifiers. We believe that this works well due to the constant repetition of vocabulary, particularly with regards to NEWS. This performance can be upheld as long as the classifier is being retrained constantly. We believe that the other classifiers could do much better given larger amounts of data, and this is something we would certainly like to test in the future. For this amount of data though, we were happy with the results achieved.

One thing that we expected and turned out to be true is that combining classifiers works well to boost our accuracy. At the very least we expected a combined classifier to perform no worse than its best component classifier, since at the very least it could just learn to listen to that classifier. But, the combined classifiers (Smart and Automated) exceeded our expectations somewhat. Our hand-written logic works slightly better than the decision tree classifier that learns the logic from training data. We discuss these issues along with the merits of each of the combined approaches in section 4.2 and 4.3. Ultimately, we concluded that the Decision Tree Combined Classifier didn't simply suffer from picking poor rules that didn't generalize well, but rather a poor discretization algorithm. For that reason, in the future, this is the place we would begin in order to improve our Combined Classifier.

6.2 Future Work

Future work would include adding more categories, which would help some classifiers further distinguish features. Examples of good categories to pursue next include real-time concerts and posts about music in general. It could also be useful to Twitter readers to differentiate another user as one who often posts factual events and news articles compared with one who offers their own opinions and critiques on topics. We believe that varying linguistic structural features and vocabulary sets could allow for such a binary classification.

Although our training and test sets were insufficiently large to necessitate this, we believe that preprocessing tweets would significantly boost scores by smoothing any inconsistencies with certain features (Go, et al.). For example, any terms such as "awesommeeeee" would be shortened to its proper English form "awesome". In partial response to the issue of global context knowledge, it is also important - particularly for the vocabulary-based classifiers - to group certain terms together; for example, we would change "oil spill" or "#BP spill" to "oilspill", recognizing these unofficial labels as a single feature.

In addition, with respect to the Combined Classifier, we discussed already in Sections 4.2, 4.3, and 6.1 that the Automated Combined Classifier could have performed much better (even better than the Smart Combined Classifier with hand-written rules) had the Decision Tree discretization algorithm been better, either by choosing more buckets and letting the DTree building algorithm do most of the work in determining where to split, or by allowing us to manually select thresholds to serve as buckets along the continuous probability space (which was the feature used). With respect to improving the Combined Classifier, this is the approach we would take.

Further work that may not be as NLP-oriented include using the other data provided by Twitter, such as the author of a Tweet (utilizing likelihood of a user to tweet in a certain category), date and time, the method by which the Tweet was posted, and regional awareness, perhaps using geolocation or simply hashtags.

7. Code Sources and References

For most of the parsers, we used code from previous CS224N assignments and the CS224N Java libraries. The N-Gram Language Model code was based on Tony and Ivan's submission for Project 1. The MaxEnt Classifier was based on Matt's submission for Project 3.

For the XML parser, we used the Apache Commons' Lang library, which has simple methods for common web data tasks such as decoding HTML entities.

In addition part of the Decision Tree code was taken from a library jaDTi.

http://www.run.montefiore.ulg.ac.be/~francois/software/jaDTi/.

The Naive Bayes classifier was based on a C# implementation found here:

http://www.codeproject.com/KB/cs/BayesClassifier.aspx

We looked at Twitter-related projects from previous offerings of CS224N, as inspiration for our projects and also to make sure our project was sufficiently unique:

Alec Go, Richa Bhayani, Lei Huang. *Twitter Sentiment Classification using Distant Supervision*.

www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf

Ravi Parikh and Matin Movassate. *Sentiment Analysis of User-Generated Twitter Updates using Various Classification Techniques*.

http://nlp.stanford.edu/courses/cs224n/2009/fp/19.pdf

John Lewin, Alex Pribula. *Extracting emotion from twitter*.

http://nlp.stanford.edu/courses/cs224n/2009/fp/22.pdf

Xavier Falco, Rafi Witten, Robin Zhou. Sentiment and Objectivity Classification.

http://nlp.stanford.edu/courses/cs224n/2009/fp/24.pdf

Suggested by course staff:

Daniel Ramage, Susan Dumais, Dan Liebling. *Characterizing Microblogs with Topic Models.*
http://nlp.stanford.edu/pubs/twitter-

icwsm10.pdf

Google Scholar:

Saˇsa Petrovic, Miles Osborne, Victor Lavrenko. *Streaming First Story Detection with application to Twitter.*

aclweb.org/anthology-new/N/N10/N10-1021.pdf

Kirill Kireyev, Leysia Palen, Kenneth M. Anderson.

Applications of Topics Models to Analysis of Disaster-Related Twitter Data

http://www.umiacs.umd.edu/~jbg/nips_tm_workshop/15.pdf

Dmitry Davidov, Oren Tsur, Ari Rappoport.

Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon

http://www.cse.huji.ac.il/~arir/10-sarcastic-twitter-conll-2010.pdf

Team Responsibilities

Ivan downloaded and labeled all of the data we used from multiple sources. He also performed research on other Twitter-based publications, wrote the Naive Bayes classifier, and wrote the code to classify user Twitter streams.

Tony worked on the N-Gram classifier and Link Domain Classifier, Decision Tree Classifier, built the N-fold validator and went on to join the classifiers with the two Combined Classifiers (Smart and Automated).

Matt created an XML parser based on the Java SAX libraries, adapted the MaxEnt classifier for Tweets, and wrote feature extractors for the tweets, which were used by the MaxEnt and Tree classifiers.