CAPSTONE PROJECT 2: IDENTIFYING SARCASM ON TWITTER

1 PROBLEM DEFINITION

Individuals often post sarcastic or ironic messages on social media sites, and businesses and government agencies need to be able to distinguish between what is intended literally and what is not. The model generated for this project enables organizations to perform this classification.

2 CLIENT DESCRIPTION

Businesses and government organizations that use text data (particularly Twitter data) will benefit from this project. For example, the Secret Service previously solicited bids for software that could distinguish valid terrorist threats from sarcastic ones (Hannon, 2014). The model generated by this project will allow such organizations to identify unlabeled sarcastic tweets and distinguish them from literal ones. This model could also aid in sentiment analysis, where determining whether a statement is intended to be sarcastic is critical for accurate classifications (Filatova, 2012; Maynard & Greenwood, 2014).

3 DATA OVERVIEW

The tweets for this project were gathered between July and October of 2017 via the Twitter REST API. The most recent 3,200 tweets made by the 3,573,510 followers of John Oliver were acquired (over 2 billion total tweets). Tweets that were retweets were excluded. Because sarcasm is difficult to identify even by human annotators (González-Ibáñez, Muresan, & Wacholder, 2011), the sarcasm hashtag is used as the true label. Of the total acquired tweets, 30,910 contain the sarcasm hashtag. The negative examples consist of hashtags with the "happy" hashtag (12,639 tweets), "sad" hashtag (40,861 tweets), and the "seriously" hashtag (11,450 tweets). In addition, profile information was gathered for each user, including number of friends, number of followers, date account was created, and location.

4 DATA WRANGLING

The following tweets were removed from the dataset:

- Duplicate tweets (tweets that appeared more than once in the dataset)
- Tweets that only contained a URL
- Tweets containing more than one hashtag (to make the comparison between sarcastic and non-sarcastic tweets clearer)
- Tweets where the hashtag of interest (#sarcasm, #happy, #sad, or #seriously) was *not* at the end of the tweet text, in line with work done by González-Ibáñez, Muresan, and Wacholder (2011)

In addition, the following data cleaning steps were performed:

- The hashtag itself was removed from the tweet text
- URLs (when present) were removed from the tweet text

5 EXPLORATORY DATA ANALYSIS

A. DATA OVERVIEW

A summary of the data collected for each hashtag is shown in Table 1. As the table indicates, the #sarcasm and #sad hashtags represent the largest portion of the tweets. There is an approximately equal number of #happy and #seriously hashtags. Similarly, the #sarcasm and #sad hashtags represent the largest portion of unique users.

Table 1. Summary of data collected

	Total Tweets	Unique Users
#sarcasm	30,910	23,509
#happy	12,639	10,149
#sad	40,861	26,456
#seriously	11,450	9,033

B. FRIENDS, FOLLOWERS, AND TOTAL TWEETS

A summary of the users' total followers, friends, and tweets is shown in Table 2. On average, regardless of the hashtag, users have more followers than friends. This indicates that the users tend to follow fewer people than follow them. As the table also shows, the users of each hashtag have a significantly different number of followers, $F(3,95139) = 4.78$, $p < .01$. Tukey's multiple comparisons test reveals that this difference is primarily driven by the fact that the number of followers for #sad and #sarcasm is significantly different ($p < .05$). The users of each hashtag also have a significantly different number of friends, $F(3, 95139) = 14.74$, $p < .001$. Here, the difference is driven by the fact that users of #happy have more followers than users of #seriously, and users of #sad have more followers than both users of #sarcasm and users of #seriously (ps $< .05$). Finally, there is a significant different in the number of total tweets users of each hashtag produced, $F(1, 95139) = 153.42$, $p < .0001$. Here, all pairwise comparisons are significant, except for #happy vs. #seriously, which are not significantly different.

Table 2. Summary statistics for friends, followers, and total tweets by hashtag

		Followers	Friends	Total Tweets
#sarcasm	Mean (SD)	833 (30,762)	646 (2,305)	4,503 (8,753)
	Range	0 - 5,192,273	0 - 76,631	2 - 446,022
#happy	Mean (SD)	826 (10,038)	711 (3,617)	3,885 (5,567)
	Range	0 - 725,605	0 - 223,235	1 - 175,701
#sad	Mean (SD)	1,227 (24,606)	772 (3,451)	5,015 (10,008)

	Range	0 - 2,136,334	0 - 485,012	2 - 384,987
#seriously	Mean (SD)	755 (10,322)	616 (951)	3,851 (6,310)
	Range	0 - 740,765	0 - 54,592	3 - 222,174

C. FREQUENCY OF HASHTAG USE

As shown in Table 3, the majority of users only produced each hashtag once. Interestingly, #sad had the highest rate of more than five uses, which isn't too surprising given that it was the most frequent of the hashtags examined here. That is, given its popularity compared to the other hashtags, it makes sense that it would be used more often in the tweets sampled.

Table 3. Percentage of users that used each hashtag X times

	#sarcasm	**#happy**	**#sad**	**#seriously**
X=1	62.16	67.72	51.95	67.18
X=2	9.33	8.79	7.46	8.24
X=3	2.50	2.27	2.44	1.95
X=4	0.93	0.81	0.90	0.76
X=5	0.51	0.32	0.54	0.24
X > 5	0.62	0.38	1.46	0.52

6 MODELING

A. SETUP

A subset of the data was selected for modeling, so that tweets with #sarcasm represented approximately 50% of the dataset. (In other words, the dataset contained an equal number of positive and negative examples.) The number and proportion of tweets with each hashtag in this dataset is shown in Table 4. While all of the available tweets with #sarcasm were used, only a subset of tweets with the other hashtags were used; these subsets were selected randomly.

Table 4. Number and proportion of tweets with each hashtag used for modeling

	Total Tweets	**Percentage of Tweets**
#sarcasm	30,728	50.14%
#happy	10,040	16.38%

#sad	10,271	16.76%
#seriously	10,262	16.74%

The data were divided into training and test sets (70-30% split), with approximately equal proportions of #sarcasm in each (50.15% of tweets in training set, 50.07% tweets in test set). For the remainder of this document, "corpus" refers to all of the tweets in the modeling dataset and "vocabulary" refers to all of the words used in the corpus.

B. BASELINE MODEL

A term-document matrix was created to represent the tweets with a vector space model. A bag of words representation was then generated using scikit-learn's CountVectorizer. Stop words, as defined by CountVectorizer, were excluded from the matrix. To identify the optimal hyperparameters, 5-fold cross-validation was performed using grid search and AUC as the scoring function. Words that occurred in fewer than two tweets were excluded, as were words that occurred in more than 70% of the tweets. Finally, a Naïve Bayes classifier was used to classify the tweets as sarcastic or non-sarcastic. Again, 5-fold cross-validation was performed using grid search and AUC as the scoring function. The model results are presented in line 1 of Table 5.

C. NAÏVE BAYES

The first model tested was a Naïve Bayes model. The procedures were the same as described above for the baseline model, except that instead of using term occurrences, I used term frequencies. Specifically, using scikit-learn's TfidfVectorizer, I transformed the count-matrix to a TF-IDF representation. Using a TF-IDF representation reduces the influence of less-informative words that occur in many tweets. The model results are presented in line 2 of Table 5. The model performance is not very different from the baseline model performance. Interestingly, both Naïve Bayes models show high recall (though not high precision), indicating that the models identified most cases of actual sarcasm (true positives), but also incorrectly labeled many non-sarcastic tweets as sarcastic (false negatives).

D. LATENT SEMANTIC INDEXING (LSI)

For the final models, LSI was used to perform dimensionality reduction and to generate "topics" by grouping together similar words. First, the vocabulary for the corpus was generated, filtering out words that occurred in fewer than 10 documents or more than 40% of the documents. Next, a bag of words representation was generated using gensim's doc2bow. Three LSI models were then generated with gensim's LsiModel, reducing the data to (a) 200 components or "topics", (b) 300 components, and (c) 400 components.

E. MODEL PERFORMANCE

Three algorithms were generated and tested using the LSI components: Random Forest, XGB Classifier, and logistic regression. As above, 5-fold cross-validation was performed using grid search and AUC as the scoring function.

As shown in Table 5, both the Random Forest and logistic regression models outperformed the Naïve Bayes models in terms of both AUC and test accuracy. However, the Random Forest models show signs of over-fitting (the training accuracy is quite high), and therefore might not be particularly useful. Overall, the logistic regression model with 400 LSI components performed the best, with an AUC of 0.74 and a test accuracy of 67.7%. The XGB models show little improvement over the Naïve Bayes models.

Table 5. Summary of model performance

Model	Feature Vectors	AUC	Precision	Recall	F1 Score	Accuracy (Train)	Accuracy (Test)
Naïve Bayes	Term occurrences	0.59	0.53	0.89	0.66	57.05%	54.12%
Naïve Bayes	Term frequencies	0.57	0.52	0.92	0.66	54.97%	53.41%
Random Forest	200 LSI components	0.67	0.63	0.60	0.62	99.70%	62.57%
Random Forest	300 LSI components	0.65	0.62	0.57	0.59	99.71%	60.80%
Random Forest	400 LSI components	0.65	0.61	0.57	0.59	99.71%	60.78%
XGB Classifier	200 LSI components	0.55	0.53	0.88	0.66	55.97%	54.81%
XGB Classifier	300 LSI components	0.57	0.54	0.72	0.62	57.24%	55.75%
XGB Classifier	400 LSI components	0.61	0.57	0.67	0.61	59.62%	57.86%
Logistic Regression	200 LSI components	0.72	0.67	0.63	0.65	66.15%	66.95%
Logistic Regression	300 LSI components	0.74	0.68	0.65	0.66	69.13%	67.33%
Logistic Regression	400 LSI components	0.74	0.68	0.65	0.67	70.04%	67.70%

7 FUTURE WORK

Future work should expand the negative examples in the dataset to tweets beyond those that use #happy, #sad, and #seriously. This would make the results more generalizable and would likely improve model performance. The model's generalizability would also be aided by training with a more random sample of tweets, since it's possible that the followers of John Oliver use sarcasm differently from Twitter users more broadly. In addition, it would be useful to adjust the proportion of positive examples (#sarcasm) and negative examples so that there are much fewer positive examples. This would more accurately reflect the distribution of tweets in the real world, where #sarcasm is only used on rare occasions.

8 REFERENCES

Filatova, E. (2012). Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In N. Calzolari, K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation* (pp. 392-398).

González-Ibáñez, R., Muresan, S., & Wacholder, N. (2011). Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Portland, Oregon: Association for Computational Linguistics.

Hannon, E. (2014, June 3). Secret Service is buying sarcasm-detecting software to know whether it thinks that last Tweet was funny. *Slate*. Retrieved from www.slate.com/blogs/the_slatest/2014/06/03/secret_service_is_buying_sarcasm_detecting_software.html

Maynard, D., & Greenwood, M. A. (2014) Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (pp. 4238-4243).