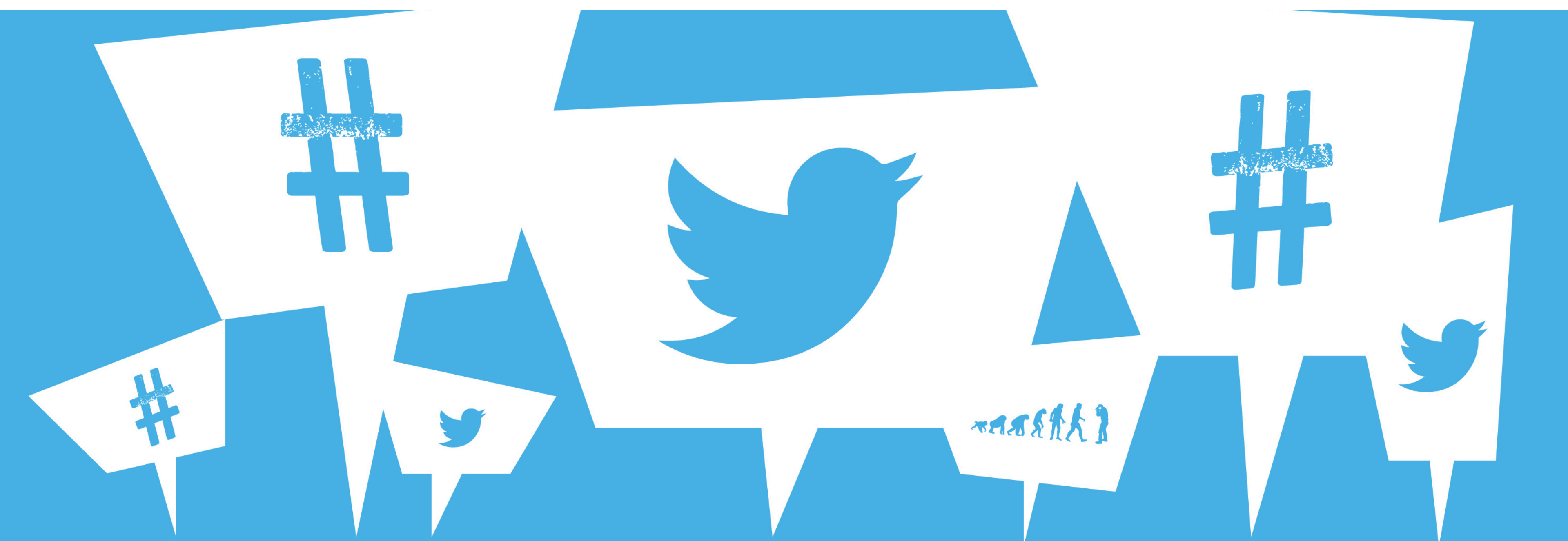
IDENTIFYING SARCASM ON TWITTER

Rachel M. Adler



PROBLEM DEFINITION

- Businesses and government organizations often need to be able to identify sarcasm online (e.g., the Secret Service[1])
- Identifying sarcasm is also vital to accurate sentiment analysis[2]
- However, determining whether a message is sarcastic is difficult for both people and computers[3]

[1] Hannon et al., 2004 [2] Filatova, 2012; Maynard & Greenwood, 2014 [3] Wallace et al., 2014

GOAL

Develop a model using Twitter data to predict whether a message is sarcastic

THE DATA

- Data collected via Twitter REST API between July and October, 2017
- Gathered most recent 3,200 tweets made by each of John Oliver’s 3,573,510 followers
- Used tweets with 4 hashtags to build models (#sarcasm, #happy, #sad, #seriously)
- Use sarcasm hashtag (*#sarcasm*) as true label

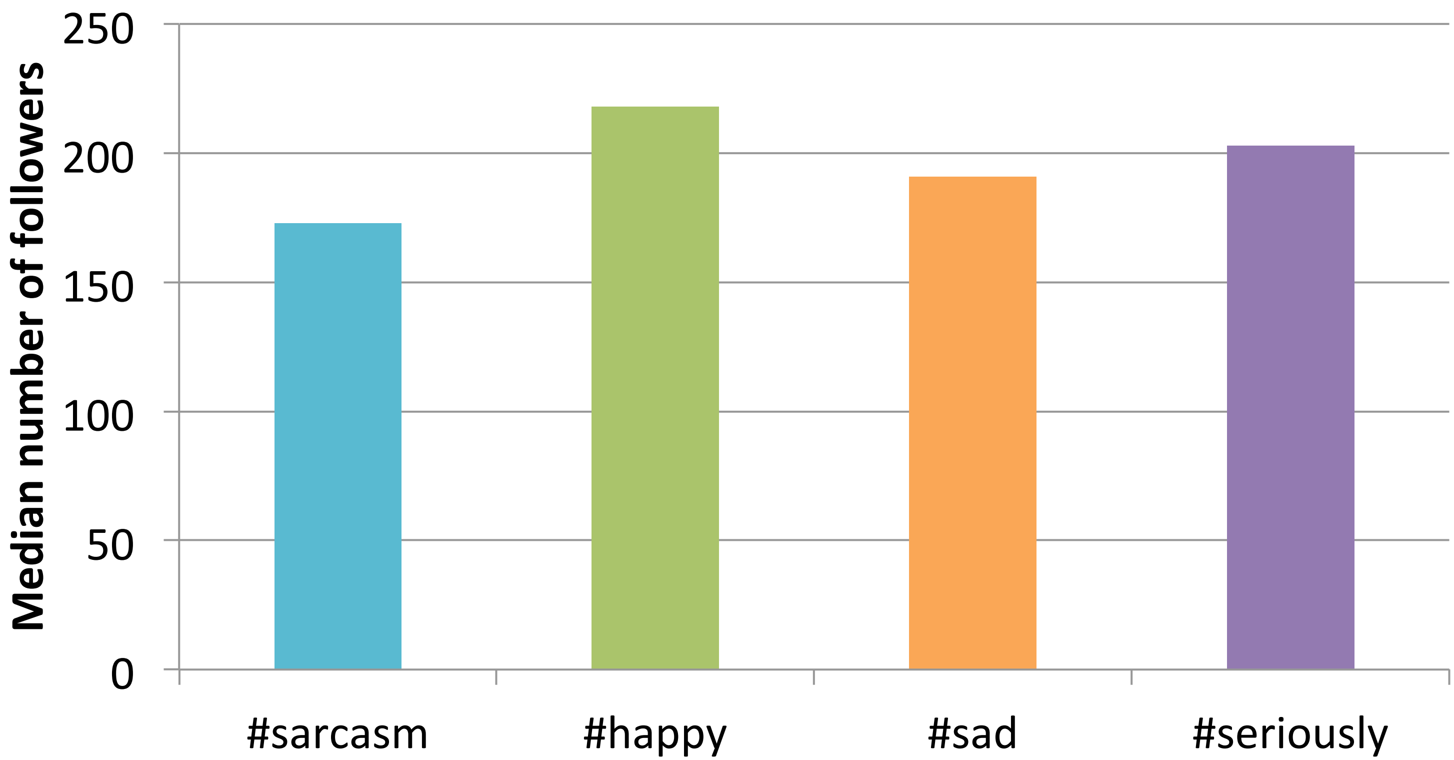
THE DATA

Summary of tweets acquired

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

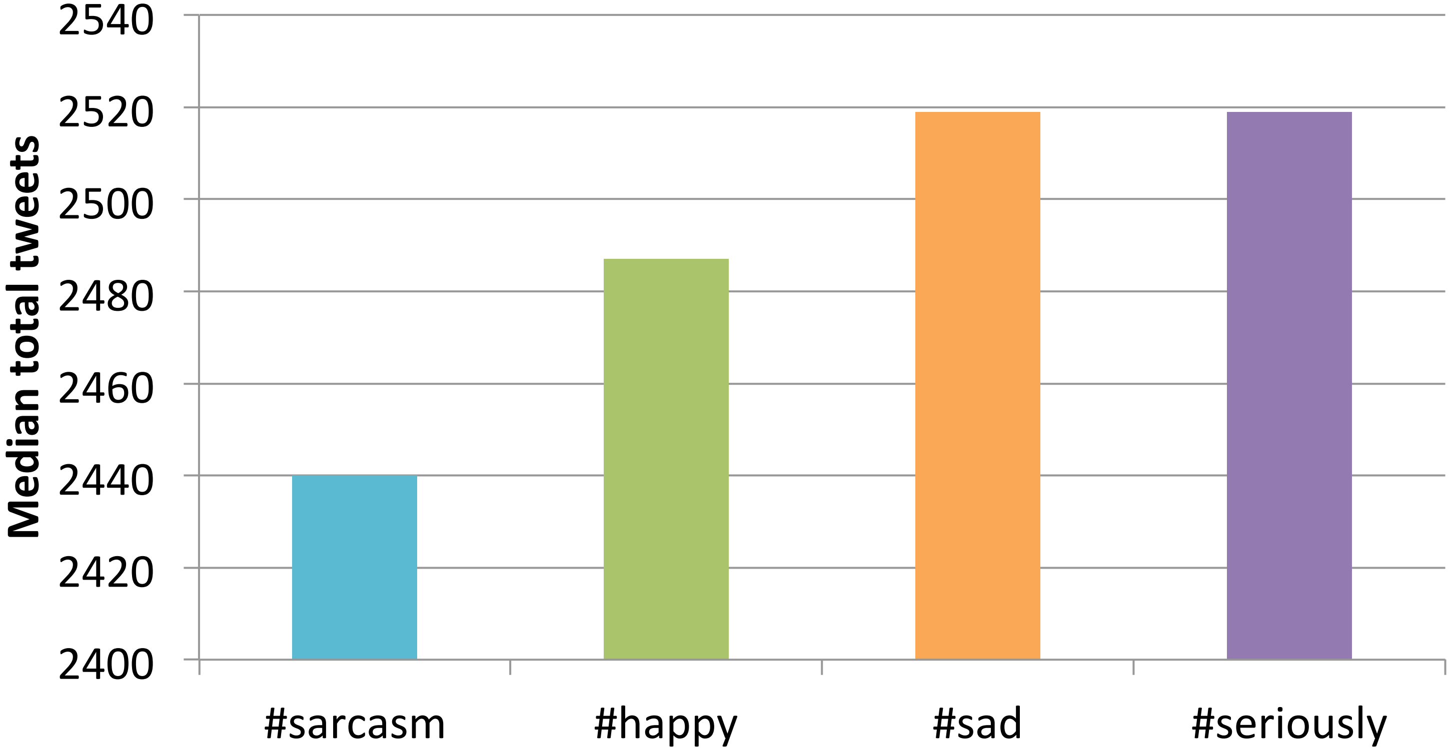
EXPLORATORY DATA ANALYSIS

Median number of followers per user



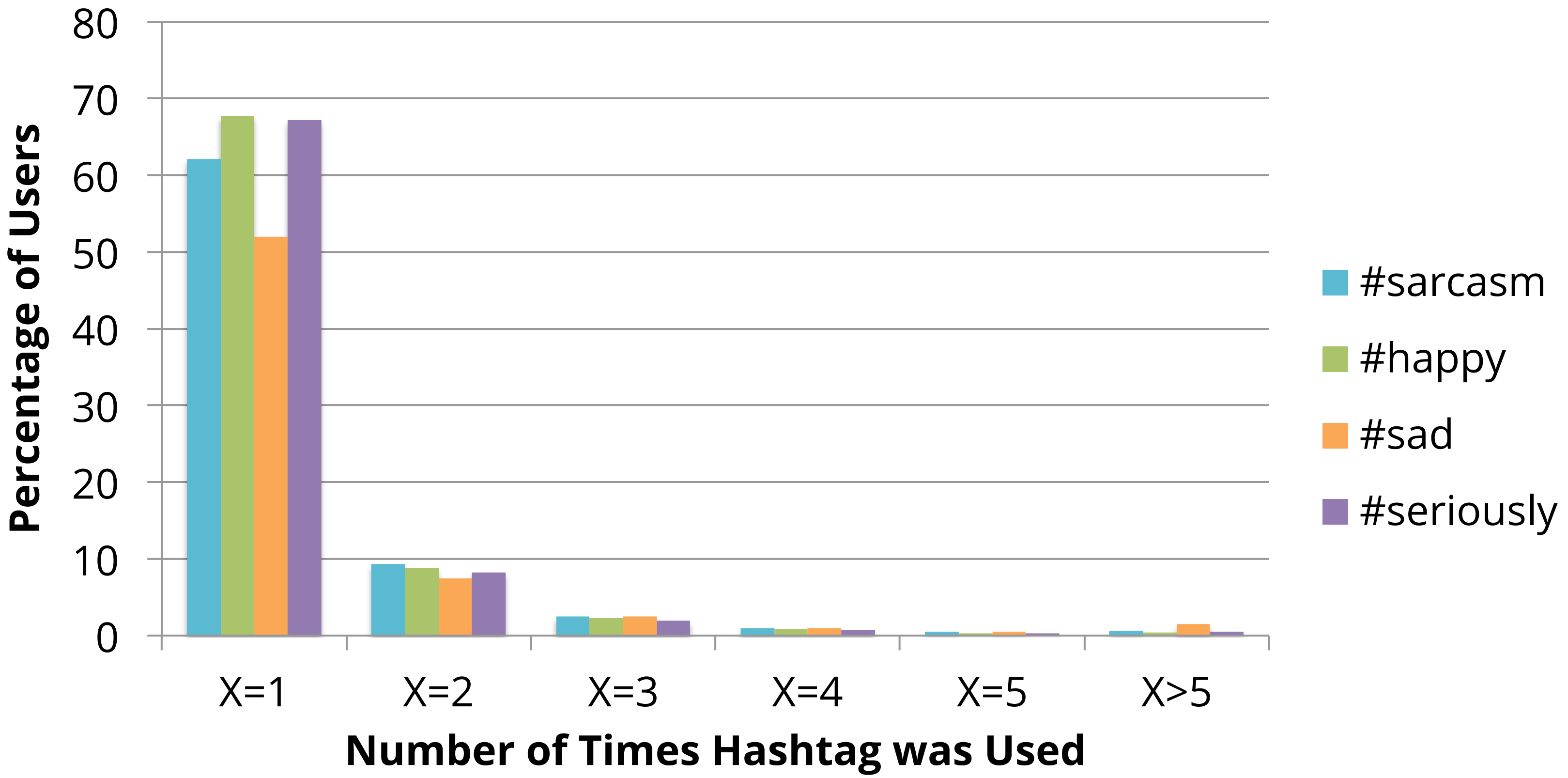
EXPLORATORY DATA ANALYSIS

Median total tweets per user



EXPLORATORY DATA ANALYSIS

Percentage of users that used each hashtag X times



MODELING SETUP

- Selected subset of data for modeling so that tweets with #sarcasm represented about 50% of dataset

	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%

- Divided data into training and test sets (70-30% split) with approximately equal proportions of #sarcasm in each

BASELINE MODEL

- Created term-document matrix to represent tweets
- Generated bag of words representation, removing stop words, with scikit-learn's *CountVectorizer*
- Performed 5-fold cross-validation with AUC as scoring function
- Used Naïve Bayes classifier; see results below

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 MODEL

- Created term-document matrix to represent tweets
- Transformed matrix to TF-IDF representation
- Generated bag of words representation, removing stop words, with scikit-learn's *TfidfVectorizer*
- Performed 5-fold cross-validation with AUC as scoring function
- Used Naïve Bayes classifier; see results below

Model	Feature Vectors	AUC	Precision	Recall	F1 Score	Accuracy (Train)	Accuracy (Test)
Naïve Bayes	Term frequencies	0.57	0.52	0.92	0.66	54.97%	53.41%

LATENT SEMANTIC INDEXING

- Used LSI to perform dimensionality reduction and to generate topics as follows:
  1. Created vocabulary for corpus
     - Excluded words appearing in fewer than 10 or more than 40% of tweets
  2. Generated bag of words representation with gensim's *doc2bow*
  3. Created three LSI models with gensim's *LsiModel*
     - 200 components
     - 300 components
     - 400 components

LSI MODELS

- Tested 3 classifiers
  - Random Forest
  - XGB Classifier
  - Logistic regression
- Tested each classifier with three LSI models (200, 300, and 400 components)
- For each model, performed 5-fold cross-validation with AUC as scoring function

RANDOM FOREST MODELS

- Table below summarizes model performance
- While AUC is higher than baseline model (AUC=0.59), high training accuracy suggests over-fitting

Model	Feature Vectors	AUC	Precision	Recall	F1 Score	Accuracy (Train)	Accuracy (Test)
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 MODELS

- Table below summarizes model performance
- Models with 200 and 300 components are worse than baseline model (AUC=0.59), though model performs better with 400 components

Model	Feature Vectors	AUC	Precision	Recall	F1 Score	Accuracy (Train)	Accuracy (Test)
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 MODELS

- Table below summarizes model performance
- All models outperform baseline model
- Model with 400 components performs best

Model	Feature Vectors	AUC	Precision	Recall	F1 Score	Accuracy (Train)	Accuracy (Test)
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%

RECOMMENDATIONS

Overall, the logistic regression model with 400 LSI components performed best and should be used for classifying sarcastic tweets

FUTURE WORK

- To make model more generalizable:
  - Expand negative examples in dataset to tweets beyond those using #happy, #sad, or #seriously
  - Use a more randomly-selected sample of tweets (rather than just tweets made by John Oliver's followers)
- Adjust proportion of positive and negative examples so there are much fewer positive examples (to better reflect distribution of tweets in real world)