Don't Take This Personally: Sentiment Analysis for Identification of "Subtweeting" on Twitter

A Senior Project submitted to
The Division of Science, Mathematics, and Computing
of
Bard College

by
Noah Segal-Gould

Annandale-on-Hudson, New York
May, 2018

Abstract

The purpose of this project is to identify subtweets. The Oxford English Dictionary defines "subtweet" as a "[Twitter post] that refers to a particular user without directly mentioning them, typically as a form of furtive mockery or criticism." This paper details a process for gathering a labeled ground truth dataset, training a classifier, and creating a Twitter bot which interacts with subtweets in real time. The Naive Bayes classifier trained in this project classifies tweets as subtweets and non-subtweets with an average F_1 score of 69%.

Contents

Dedication

I dedicate this senior project to @jack, who has willfully made numerous changes to Twitter which inevitably angered millions.

Acknowledgments

Thank you professors Sven Anderson, Keith O'Hara, and Rebecca Thomas for making this project possible through your combined efforts to teach and advise me. Thank you Benjamin Sernau '17 for enduring through three years of Computer Science courses with me and being a source of unending joy in my life. Thank you to Julia Berry '18, Aaron Krapf '18, and Zoe Terhune '18 for being my very best friends and giving me things worth caring about. Finally, thank you to my parents Tammy Segal and Emily Taylor for your constant support and patience throughout my four years at Bard College.

1
Introduction

1.1 Background

The news and social networking service Twitter had over 140 million active users who sent 340 million text-based Tweets to the platform every day by March of 2012 [1]. Since Twitter-founder Jack Dorsey sent the first Tweet in March of 2006 [2] social scientists, political scientists, and computer scientists have applied machine learning techniques to understand the patterns and structures of the conversations held on the platform. One such technique is sentiment analysis, which seeks to ascertain the opinions of bodies of text. Sentiment analysis techniques are often treated as classification problems which seek to place text into categories such as **positive**, **negative**, and **neutral**.

On Twitter, the most common way to publicly communicate with another user is to compose a tweet and place an "@" before the username of that user somewhere in the tweet (e.g. "How are you doing, @NoahSegalGould?"). Through this method, public discussions on Twitter maintain a kind of accountability: even if one were to miss the notification that they were mentioned in a tweet, one's own dashboard keeps a running list of their most recent mentions.

If an individual sought to disparage or mock another, they could certainly do so directly. But the targeted user would probably notice, and through the search functions of the platform, anyone could see who has mentioned either their own or another's username. Instead, a phenomenon

persists in which users of the platform deliberately insult others in the vaguest way possible. Tweets of this kind are colloquially called "subtweets" and typically target a specific person but do not contain the username of that person.

All users do not necessarily possess the same exact definition of "subtweet." I trust the Oxford English Dictionary's "[tweet] that refers to a particular user without directly mentioning them, typically as a form of furtive mockery or criticism," however that definition is perhaps too restrictive. Some individuals believe subtweets abide by this definition, but others expand it to allow inclusion of others' real names (especially if that individual does not own a Twitter account), and some do not even require that a particular user be the target of the tweet. In this project, I implement a classifier which abides by a particularly loose definition in order to please as many parties as possible.

1.2 Motivation

The inspiration for this project came from interests I garnered taking courses within Bard College's Computer Science department as well as its Experimental Humanities concentration. The very first course I attended at Bard College was Professor Keith O'Hara's *Object-Oriented Programming with Robots*. It served as my first introduction to computer programming, and for my final project I created *Fuzzfeed*: a Twitter bot which generated fake *Buzzfeed* article titles. The following academic year, I wrote programs in Professor Collin Jennings' *Signs and Symbols: Patter Recognition in Literature and Code* and Professor Rune Olsen's *Cybergraphics* which analyzed and visualized topic models of poetry on Twitter. The first time I implemented sentiment analysis on my own was in Professor Gretta Tritch-Roman's *Mapping the 19th Century City*, for which I sought to analyze 1860s New York City newspapers for their sentiments toward immigration. Most recently, I utilized K-Means clustering and topic modeling on tweets written by members of US congress in Professor Anderson's *Introduction to Artificial Intelligence* course.

By my Junior year, my friends and I used Twitter on a daily basis. In my free time, I made Twitter bots that utilized Markov chains to generate text based on corpora of their tweets. Data

collection became a passion of mine as I learned to appreciate the utility and acknowledge the deliberate limitations of web-based APIs. I taught myself web-scraping and utilized Python's *BeautifulSoup* library to programmatically acquire text from Bard College's official online course catalog as well as Twitter's web interface. These skills for programmatically interacting with the world wide web became useful resources during the completion of this project.

My peers introduced me to subtweeting, and I started to pay closer attention to tweets that followed the typical patterns of distanced criticism that subtweets were known for. Because some format seemed to exist which was popularly applied to produce the optimal subtweet, I pitched the concept of subtweet classification to my senior project adviser, Professor Sven Anderson, and I started work on this project in the Fall.

I was initially motivated to complete a senior project on this topic because I wanted to create something useful to my peers and also challenge their notions of public and private interactions on social networking applications like Twitter. Individuals I knew personally would take to the platform to complain indirectly about one another through their subtweets. Friends and I shared evenings debating on if a particular mutual friend's complaints were actually subtweets, and I wondered if that guess-work could be done by a program. I also wanted to challenge the hypocrisy of utilizing a service which presents itself as a public forum to speak in distinctly private ways. Toward this end, I decided the project would be in pursuit of the following goals: it would provide a framework for collecting examples of subtweets, train a classification algorithm using those examples, and finally utilize that classifier in real time to make tweets which were intended to be unseen by specific parties easily accessible to all parties. In presenting covertly hurtful content as obviously hurtful in a public fashion, perhaps I could promote a particular awareness that tweets posted by public accounts were indeed publicly accessible, and that Twitter's Terms of Service [3] allowed for this kind of monitoring.

1.3 Literature Review

Long before Twitter, psychologist Gordon Allport wrote about "antilocution" in *The Nature of Prejudice* [4]. For Allport, antilocution was the first of several degrees of apathy which measure prejudice in a society and represented the kind of remarks which target a person, group, or community in a public or private setting but do not address the targeted individual directly. Different from both hate speech and subtweeting, antilocution necessitates that an in-group ostracize an unaware out-group through its biases.

"Subtweet" was coined in December of 2009 by Twitter user Chelsea Rae [5] and was entered into Urban Dictionary the following August [6]. In "To tweet or subtweet?: Impacts of social networking post directness and valence on interpersonal impressions" [7], Edwards and Harris sought to analyze student participants' perceptions of known subtweeters. In the news, too, subtweets have garnered attention in *The Atlantic* [8], *The Washington Post* [9], and *Slate* [10]. In news media, subtweets garner attention for their prevalence among government officials as well. Following President Donald Trump's inauguration, The Washington Post compiled its "A running list of all the possible subtweets of President Trump from government Twitter accounts," [11] cementing subtweets as particularly newsworthy.

Instead of focus groups, opinion polls, and conduct surveys, sentiment analysis and opinion mining programs are increasingly applied to social networking websites to analyze the sentiments and opinions of users toward topics and products. For the Twitter social networking platform, sentiment analysis allows administrators to enforce their hateful conduct policies [12] which specifically prohibit violent threats and some types of degrading content.

Sentiment analysis on social networking services such as Twitter has garnered attention within seemingly distinct fields of interest. In "Text mining for market prediction: A systematic review," Nassirtoussi et al. surveyed varied methods for text-mining social media for sentiment analysis of financial markets and considered that field with both behavioral and economic considerations in mind [13]. Following a terrorist event in Woolwich, London in 2013, Burnap et al. analyzed the immediate Twitter response following the attack to inform statistics on how long it takes

for responses from official sources to disseminate during crises [14]. Prior research of these kinds utilizes sentiment analysis techniques on tweets, but no known research exists which specifically performs any sentiment analysis on subtweets.

The most germane research available focuses on sentiment analysis of figurative language. Determining sentiment based on features of text which are distinctly separate from their literal interpretations presents difficulties for human readers as well as computer programs. In *SemEval*, the International Workshop on Semantic Evaluation, analysis of figurative language on Twitter has been a core task for their competition since 2015 [15] and returns this year with a specific focus on ironic tweets [16]. In this year's description for "Task 3: Irony detection in English tweets," Van Hee et al. touch upon online harassment as a potential point of significance for sentiment analysis of ironic tweets.

Using a machine-learning approach to perform sentiment analysis, syntactic and linguistic features are typically utilized in probabilistic (e.g. Naive Bayes and Maximum Entropy) and linear (e.g. Support Vector Machines and Neural Networks) classification algorithms. The probabilistic approach is sometimes called *generative* because such models generate the probabilities of sampling particular terms [17]. Linear classification utilizes the vectorized feature space of words, sentences, or documents to find a separating hyperplane between multiple classes. Of these approaches, Naive Bayes stands out as particularly simple and common. A **bag of words** model typically ignores word positions and then Bayes Theorem is utilized to predict the probability that a given feature set (e.g. words, sentences, etc.) belongs to a particular label (i.e. a category or class). The naive assumption maintains that all features are treated as conditionally independent (i.e. that the presence or omission of a particular feature does not change the likelihood of encountering other features), and although this is frequently violated, Naive Bayes often performs well anyway [18]. For cases in which the classifier encounters a feature absent from the features which were used to train it, a so-called **zero probability** appears. Because the probability of encountering the feature is 0, **additive smoothing** is often utilized to ap-

propriately weight new features so the probability of an entire feature-set fitting into a specific class is not 0.

1.4 Changes in Data Acquisition

The novel approach I developed for creating a ground truth dataset relied on a particular phenomenon in which Twitter users were already calling-out the subtweets of their peers. The format I noticed followed that a user would post a subtweet which was easily recognized by a peer, and that peer would then reply to that tweet in order to complain that the original user was subtweeting or to ask if the tweet was indeed a subtweet. Initially, the Python script I wrote to utilize the Twitter API's search functionality via the Tweepy library specifically searched for replies to tweets which contained some form of the string "subtweet" and then utilized the API's status object to access the tweet to which it was replying. Both the alleged subtweet and the tweet containing the accusation were saved to a comma-separated values (CSV) file. I ran the script every day for over two months.

Initially, I trained the classifier using a dataset which was half comprised of these alleged subtweets and half comprised of tweets randomly selected from a dataset provided by Alec Go [19]. That was a mistake. I had failed to make the training data representative of actual subtweets and non-subtweets. To rectify this, I revised the alleged subtweets downloading script and created one that had the opposite effect: it downloaded tweets with replies which specifically did **not** contain the string "subtweet." In both the script which downloaded subtweets and the script which downloaded non-subtweets, I knew my assumptions about these interactions would not hold true in every case. They were intended as generalizations which would make acquiring a ground truth dataset for use in performing binary classification significantly easier and less time-consuming than finding and labeling subtweets and non-subtweets by hand. Indeed, Alec Go's aforementioned dataset utilized a similar method for acquiring labeled data. In his *Sentiment140* dataset, the labels were acquired according to emoticons present within the tweets instead of through hand-labeling by actual humans.

1.5 The Twitter API

[Explain how it is accessed and its limitations.]

1.6 Regular Expressions, N-Grams, & Tokenization

[Explain the use for regular expressions in identifying hashtags, URLs, and mentions.]

[Explain N-Grams and how they help a classifier.]

[Explain tokenization and specifically NLTK's TweetTokenizer.]

1.7 TF & TF-IDF

TF-IDF, or term frequency-inverse document frequency, is a statistical representation of how important a single word is for each document in a collection of documents.

[Explain vectorization and provide examples.]

1.8 Naive Bayes

Naive Bayes classifiers are probabilistic supervised learning models which make the "naive" assumption of independence between pairs of features being classified. Sentiment analysis is popularly performed through Naive Bayes.

[Explain some probability, the independence assumption, and the multinomial distribution. Give examples.]

1.9 Statistical Considerations

In tasks pertaining to text classification, like sentiment analysis, precision refers to the number of correctly labeled items which were labeled as belonging to the positive class and in fact did belong to that class (true positives) divided by the total number of elements which were labeled as belonging to the positive class including ones which were labeled positively either correctly

or incorrectly. Recall, then, refers to the true positives divided by the total number of elements that actually belong to the positive class.

[Explain F1, Precision, Recall, Accuracy, and Null Accuracy.]

2
Implementation

2.1 Searching for Tweets Using the Twitter API

[Show code and explain how searching works with Tweepy.]

2.2 Cleaning the Data

[Show code and explain how text cleaning genericizes certain features and ignores tweets lacking enough English words.]

2.3 Training the Classifier & K-Folds Cross-Validation

[Explain how pipelines are trained and how K-Folds splits the dataset.]

3
Results

3.1 Ground Truth Dataset

[Explain the tables and figures.]

3.2 Confusion Matrices

A confusion matrix is a table which visualizes the performance of an algorithm. In this case, I implemented a Naive Bayes classifier from Scikit Learn on my dataset and included in my results is a confusion matrix of the performance...

[Explain how to read a confusion matrix and show the test and training figures.]

3.3 Statistical Analyses

3.4 Application of the Classifier on Tweets from Known Subtweeters

3.5 Most Informative Features

[Explain how to read the most informative features for each class (or just the "subtweet" class) and show the table.]

[Show the scores from K-Folds.]

3.6 The Twitter Bot

[Explain how the Twitter bot works and show code as well as examples of interactions.]

3.7 Discussion

[Discussion of results.]

4
Conclusion

4.1 Summary of Project Achievements

[Ground truth data, classifier, cleaning, Twitter bot.]

4.2 Future Work & Considerations

[Test other classification algorithms. Utilize more training data.]

Bibliography

[1] Twitter, *Twitter turns six* (2012), available at https://blog.twitter.com/official/en_us/a/2012/twitter-turns-six.html.

[2] Jack Dorsey, *inviting coworkers* (2006), available at https://twitter.com/jack/status/29.

[3] Twitter, *Twitter Terms of Service* (2016), available at https://twitter.com/en/tos.

[4] Gordon W Allport, *The nature of prejudice* (1954).

[5] Chelsea Rae, *I hate when i see people...* (2009), available at https://twitter.com/Chelsea_x_Rae/status/6261479092.

[6] Urban Dictionary, *Subtweet* (2010), available at https://www.urbandictionary.com/define.php?term=subtweet.

[7] Autumn Edwards and Christina J Harris, *To tweet or subtweet?: Impacts of social networking post directness and valence on interpersonal impressions*, Computers in Human Behavior **63** (2016), 304–310.

[8] Alexis C. Madrigal, *Behind the machine's back: How social media users avoid getting turned into big data* (2014), available at https://goo.gl/h36jxx.

[9] Caitlin Dewey, *Study confirms what you always knew: People who subtweet are terrible* (2016), available at https://goo.gl/SeV3mx.

[10] Chelsea Hassler, *Subtweeting looks terrible on you. (you know who you are.)* (2016), available at https://goo.gl/NCz27z.

[11] Abby Ohlheiser, *A running list of all the possible subtweets of President Trump from government Twitter accounts* (2017), available at https://goo.gl/hFh81R.

[12] Twitter, *Hateful conduct policy* (2018), available at https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy.

[13] Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo, *Text mining for market prediction: A systematic review*, Expert Systems with Applications **41** (2014), no. 16, 7653–7670.

[14] Pete Burnap, Matthew L Williams, Luke Sloan, Omer Rana, William Housley, Adam Edwards, Vincent Knight, Rob Procter, and Alex Voss, *Tweeting the terror: modelling the social media reaction to the woolwich terrorist attack*, Social Network Analysis and Mining **4** (2014), no. 1, 206.

[15] Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes, *Semeval-2015 task 11: Sentiment analysis of figurative language in twitter*, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (2015), 470–478.

[16] Cynthia Van Hee, Els Lefever, and Vronique Host, *Semeval-2018 Task 3: Irony detection in English Tweets*, Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018) (2018).

[17] Walaa Medhat, Ahmed Hassan, and Hoda Korashy, *Sentiment analysis algorithms and applications: A survey*, Ain Shams Engineering Journal **5** (2014), no. 4, 1093–1113.

[18] Harry Zhang, *The optimality of naive Bayes*, AA **1** (2004), no. 2, 3.

[19] Alec Go, Richa Bhayani, and Lei Huang, *Twitter Sentiment Classification using Distant Supervision*, CS224N Project Report, Stanford (2009), 12.