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 71%.

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

The data gathered for this project contains unsavory language which may be considered inappropriate.

1.1 Background

This project utilizes the Twitter Application Programming Interface (API) as well as concepts from machine learning and text analysis. The ground truth dataset created for this project features 30,564 tweets by 27,711 unique users from March and April of 2018. There has been significant research in the field of sentiment analysis to identify the opinions held within bodies of text, however this research does not include so-called "subtweets." The Oxford English Dictionary defines that a subtweet is a "[tweet] that refers to a particular user without directly mentioning them, typically as a form of furtive mockery or criticism." Treated as a characteristic of a tweet's sentiment, we will utilize the Naive Bayes classification algorithm to perform sentiment analysis on subtweets.

For acquisition of a ground truth dataset, we consider **true subtweets** to be tweets to which another Twitter user replied who specifically called it out as a subtweet. We consider **true non-subtweets** to be tweets to which another user replied who specifically did **not** call it out as a subtweet. Consider these examples:

	True Subtweet Data	True Non-Subtweet Data
Tweet	im about to start ignoring ppl as they annoy me, maybe they'll get the hint	That's been one of my biggest issues here; the onus is on ordinary people who, in their spare time, must campaign for the basic services of a city. This is not how progressive cities should be built
Reply	I'd rather subtweet you	i guess i am not as creative as i thought

We will keep these definitions and examples in mind when the classifier is used on original tweets independent of their replies. These will be called **predicted subtweets** and **predicted non-subtweets**. The following sections detail the resources and techniques utilized to acquire the labeled ground truth dataset, train the Naive Bayes classifier, and program a Twitter bot to interact with subtweets in real time.

1.2 Changes in Data Acquisition

The novel approach 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 following pattern was observed: 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 program used the Twitter API's search functionality to specifically search for replies to tweets which contained some form of the string "subtweet." It utilized the API's status object to access the tweet to which it was replying. For two months, each day's alleged subtweets and their associated accusatory replies were saved.

Initially, the classifier was trained using a dataset which was half composed of these alleged subtweets and half composed of tweets randomly selected from a pre-labeled sentiment analyzed tweets dataset (Go et al., 2009). This procedure failed to make the training data representative of **true subtweets** and **true non-subtweets**. The alleged subtweets downloading program was revised and it was set to download tweets with replies which specifically did **not** contain the

string "subtweet." In both the program which downloaded subtweets and the program which downloaded non-subtweets, the 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.

1.3 The Twitter API

Twitter provides a free Application Programming Interface (API) to registered users and has done so since September of 2006 (Stone, 2006). The API allows developers to programmatically access and influence tweets individually or through real time search filters, and also read and write direct messages (Twitter, 2018). The creation of a Twitter application which utilizes the API requires creation and email verification of an account, and developers are also required to agree to the terms of service (Twitter, 2016). Creation of an application provides developers with authentication tokens which can then be used to access the API.

To make creation of Twitter applications easier, Tweepy (Roesslein, 2009) is an open source library for the Python programming language which provides methods and classes used to interact with the API and its status objects (Twitter, 2018). A Twitter status object is a dictionary of key and value pairs which contains text, media, and user information associated with particular tweets (i.e. statuses). There are rate limits for both reading and writing to the API which must be kept in mind when programming for it.

1.4 Text Cleaning

Text cleaning is the process by which text is modified prior to any kind of processing. Changes are made to preserve the characteristics of the text which are relevant to the goal of the analysis

and to leave out the ones which are irrelevant. Because we will be using Naive Bayes, we must keep in mind which features in each tweet (e.g. URLs and user names) ought to influence the probabilities that an entire feature-set (i.e. that whole tweet) suits a particular class.

1.4.1 Regular Expressions

For text classification through machine learning, it is popular to modify the ground truth dataset to make features which are not important to the classification problem as **generic** as possible. For classification of subtweets, the classifier will treat URLs, mentions of usernames, and English first names generically. In other words, it will keep track of the existence of those features but specifically will not encounter the text contained within them. In identification of subtweets, there exists no syntactic or linguistic significance in the format of a URL or the name a user chooses to associate with themselves or another. However, the existence of those features within the tweet remains important. For this kind of substring searching, pattern matching through regular expressions was used to replace every occurence of URLs, usernames, and first names with special tokens which were not already in the dataset. The top 100 most common English names for both men and women over the last century were acquired from the United States Department of Social Security.

1.4.2 Tokenization

Instead of training the classifier on entire strings, **tokenization** is necessary in order to extract individual features from the text. The Natural Language Toolkit (NLTK) provides a tweet tokenizer to achieve this. For some string, the tokenizer splits apart words, usernames, URLs, hashtags, and punctuating characters as individual tokens. NLTK's tweet tokenizer also appropriately distinguishes between punctuating characters and emoticons composed of punctuating characters.

1.4.3 N-Grams

An **n-gram** is a contiguous sequence of n tokens in a piece of text. For example, given a string such as "This is a test," the bigrams ($n = 2$) for this string are "This is," "is a," and "a test."

Instead of training the classifier using unigrams ($n = 1$) exclusively, we train it using unigrams, bigrams, and trigrams ($n = 3$). Thus, when the probability that some specific token within a tweet belongs to a specific class is calculated, its neighbors are also considered in combination with it. N-grams enable the classifier to treat particular groupings of tokens with some size n as importantly as it treats the individual tokens, thus identifying particular word groupings most associated with the classes.

1.4.4 Stop Words

A list of stop words typically contains the most common words in a language. For English text, the list is often composed of words such as "the," "it," and "of." Tokens matching stop words are ignored during classifier training because they are too common to help the classifier distinguish subtweets from non-subtweets.

1.5 TF & TF-IDF

The probabilities calculated for Naive Bayes are not best found using raw token counts within individual documents. Instead, TF and TF-IDF are popularly used in order to vectorize the tokens for use in training the classifier.

Term frequency (TF) is a simple method for vectorizing text in which all terms (i.e. tokens, features, words, etc.) in the corpus are featured in a vector for each document, and the frequency of each term is reflected in the number representing the corresponding term. Unfortunately, TF falls short when the corpus of documents contains terms which appear frequently but do not necessarily help inform the classifier on terms that are best associated with a particular class. TF-IDF, or term frequency-inverse document frequency, is the product of the TF for a specific term and the inverse document frequency (IDF) for that same term. The TF is equal to the ratio between the number of occurrences of a term in a document, and the total number of words in that document. IDF, then, is the logarithm of the ratio between the number of documents in the corpus, and the number of documents which contain that term. Taking the logarithm means the value will be higher for rarer terms. Thus, the product of TF and IDF assigns weights which

appropriately value terms which are frequent within a document but rare in the entire corpus of documents.

1.6 Naive Bayes

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

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

1.7 Statistical Considerations

In the binary classification of subtweets and non-subtweets, we consider true positives (TP) to be true subtweets which were correctly labeled as predicted subtweets, false positives (FP) to be true non-subtweets which were incorrectly labeled as predicted subtweets, true negatives (TN) to be true non-subtweets which were correctly labeled as predicted non-subtweets, and false negatives (FN) to be true subtweets which were incorrectly labeled as predicted non-subtweets. As such, there are two ways for the classifier to be wrong: it can produce false negatives and false positives.

1.7.1 Precision

Precision refers to the ratio between the true positives, and the true positives and false positives. It is also known as the positive predictive value.

$$P = \frac{TP}{TP + FP}$$

1.7.2 Recall

Recall, then, refers to the ratio between the number of true positives, and the true positives and false negatives. It is also known as the sensitivity.

$$R = \frac{TP}{TP + FN}$$

1.7.3 Accuracy

The accuracy is the ratio between the true positives and the true negatives, and the true positives, true negatives, false positives, and false negatives. Accuracy alone is a particularly bad quantifier of how well a classifier performs when working with data which is class-imbalanced (i.e. there are not equal numbers of items in each class). In our ground truth dataset, the classes are balanced so accuracy is fine.

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

1.7.4 F1 Score

The F_1 score is a weighted average of the precision and recall. Thus, it takes both false positives and false negatives into account.

$$F1 = \frac{2 * (P * R)}{P + R}$$

1.7.5 Null Accuracy

The null accuracy is just the accuracy which is obtained by always predicting the most frequent class. Because there are two classes and the tweets within the ground truth dataset equally compose both, the null accuracy will always be 50%.

1.8 Review of Literature

"Subtweet" was coined in December of 2009 by Twitter user Chelsea Rae (Rae, 2009) and was entered into Urban Dictionary the following August (Urban Dictionary, 2010). In "To tweet or subtweet?: Impacts of social networking post directness and valence on interpersonal impressions" (Edwards and Harris, 2016), Edwards and Harris sought to analyze student participants'

perceptions of known subtweeters. In the news, too, subtweets have garnered attention in *The Atlantic* (Madrigal, 2014), *The Washington Post* (Dewey, 2016), and *Slate* (Hassler, 2016). 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," (Ohlheiser, 2017) cementing subtweets as particularly newsworthy.

There were over 140 million active Twitter users who sent 340 million text-based tweets to the platform every day by March of 2012 (Twitter, 2012). Since Twitter-founder Jack Dorsey sent the first Tweet in March of 2006 (Dorsey, 2006) 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. Through sentiment analysis, we are able to use machine learning to identify patterns within natural language which indicate particular feelings both broadly (e.g. positive, neutral, or negative) and toward topics (e.g. politics, terrorism, brands, etc.).

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 a vague manner by making complaints while omitting the targets of those complaints.

Although the OED's definition states that a subtweet "...refers to a particular user without directly mentioning them, typically as a form of furtive mockery or criticism," it 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. Because subtweeting is colloquial in nature, we will expand the definition of subtweet to permit these less restrictive features.

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 approached that problem with both behavioral and economic considerations in mind (Nassirtoussi et al., 2014). 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 (Burnap et al., 2014). Prior research of these kinds utilizes sentiment analysis techniques on tweets, but no known research exists which specifically performs any sentiment analysis on subtweets.

Long before Twitter, psychologist Gordon Allport wrote about "antilocution" in *The Nature of Prejudice* (Allport, 1954). 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.

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 (Ghosh et al., 2015) and returns this year with a specific focus on ironic tweets (Van Hee et al., 2018). 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.

We pursue sentiment analysis of subtweets in order to challenge the hypocrisy of utilizing a service which presents itself as a public forum to speak in distinctly private ways. Toward this end, these are our goals: this project will 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 it will promote a particular awareness that tweets posted by public accounts are indeed publicly accessible, and that Twitter's Terms of Service (Twitter, 2016) allows for this kind of monitoring.

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 (Medhat et al., 2014). 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 (Zhang, 2004). 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 appropriately weight new features so the probability that an entire feature-set fits into a specific class is not 0.

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

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

Dorsey, Jack. 2006. *inviting coworkers*, available at https://twitter.com/jack/status/29.

Twitter. 2016. *Twitter Terms of Service*, available at https://twitter.com/en/tos.

Allport, Gordon W. 1954. *The nature of prejudice*.

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

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

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

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

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

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

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

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

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

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

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

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

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

Zhang, Harry. 2004. *The optimality of naive Bayes*, AA **1**, no. 2, 3.

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

Stone, Biz. 2006. *Introducing the Twitter API*, available at `https://blog.twitter.com/official/en_us/a/2006/introducing-the-twitter-api.html`.

Twitter. 2018. *Twitter API Docs*, available at `https://developer.twitter.com/en/docs`.

Roesslein, Joshua. 2009. *tweepy Documentation* **5**, available at `http://docs.tweepy.org/en/v3.5.0/`.

Twitter. 2018. *Tweet objects*, available at `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object`.