Exploring the presence of discriminatory issues in Twitter.

Morgan Dock
Graduate Student
University of Mississippi

Ethan Luckett
Undergraduate Student
University of Mississippi

Dixon Styres
Graduate Student
University of Mississippi

ABSTRACT
In this paper, we explore the presence of discriminatory remarks on Twitter.

1. INTRODUCTION
Twitter is a booming social media platform that has a heavy reliance on portraying both live reaction and free speech. It's parallel attributes to actual conversations make it an excellent tool at portraying one's thoughts and ideas to the world at large. However, as observed in actual conversation, many people project ideals onto others that are considered discriminatory and derogatory to certain groups. When these ideals are applied to twitter streams of individuals they provide a quick way to publish slander to millions of people across the globe. Our project sets out to identify discriminatory speech and provide map based locations as well as data about users in these locations with a goal set at reducing the impact of discriminatory remarks to the masses. With data that has been collected and analyzed through this project we would be able to find and alter the way of highly discriminatory users of twitter reach the masses perhaps by filtering or limiting their use of the site. Goals for this project include determining where discriminatory tweets are taking place, overall sentiment for places involving certain topics, and finding top accounts of users relating to discriminatory remarks. Technically, our approach collects tweets using the Twitter streaming API and classifies them via our self-labeled discriminatory data to determine the discriminatory "type" of a given tweet. We then display the tweet on a web based GUI using the Google Maps API.

2. DATA COLLECTION
Data was collection was split into two parts, model training, and live streaming analysis. Both methods involved the use of the Twitter streaming API. For interfacing with the API, we used *Tweepy* a collection of python methods for collecting data from the streaming API. To collect data for model training, we began by creating a list of "badwords", words that we believed to be relevant to discriminatory content. We decided to label tweets based on 6 types: political, disabilities, sexual orientation, racial, gender, and religion. We then collected 1500 tweets from the stream that collected words in our list. After collection, we manually labeled the tweets to determine their class and sentiment. For the live streaming aspect of our project we used the same "badwords" list to collect relevant tweets, preprocessed them and asked our classifier to classify them. This data was then displayed onto our GUI

Table 1. Collected and labeled data

Index	Text	Type	Bottom
0	Think we should consider banning mulims after all these attacks #build thatwall.	Religious	Pro
1	Stand up for everyones Lesbian rights, today we fight for a better future!	Sexual Orientation	Anti

3. PROBLEM DEFINITION
3.1 Type Detection
Type detection involves determining if a given tweet is relevant to a certain class of discriminatory types, and to which of our 6 types it belongs. We setup a KNN classifier to handle this problem. KNN is a supervised learning algorithm that classifies input based on trained data and whose output is class membership. For input into our classifier we perform a few preprocessing tasks that helped to select relevant data and to vectorize our input. To preprocess our tweets we took several steps to ensure we were only classifying and training relevant words.

1. Lowercase conversion: as to not differentiate between WALL and wall.
2. URL Removal: We did not intend on following links and could not correlate having a link in a tweet with any known type or sentiment
3. Whitespace Cleanup: Clear extra whitespace and tabs.
4. @Usernames: We determined a username did not provide any type or sentiment data.
5. Non Letters: Numbers and other non-textual items do not contribute any sentiment or type data.
6. Two Or More Characters: At times users type extra characters to show emphasis on a word such as "looooong day" we remove two or more characters from a word to clean this up.
7. All stopwords were removed.

Objects are classified by a majority vote of its neighbors, with the object being assigned to the class most common among its neighbors. Our implementation of the KNN algorithm is concurrent with the designed algorithms present in *SciKit Learn*: a classification suite for python containing many tools for data analysis. Once the classifier trained on our labeled data, we were

able to feed it a tweet so that it may classify it based on the tweets textual data, and fit it into one of our 6 discriminatory sections. The idea is to treat each tweet as a finite mixture over a set of topics, each of which is in turn characterized by a word distribution, and then examine tweets via such topic distributions.

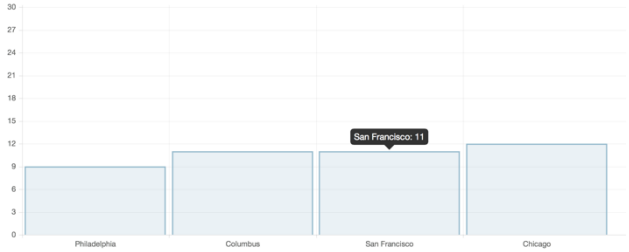
3.2 Attitude Detection

Another task in type detection was that of detecting sentiment. For this, we used a second KNN classifier and fed it the same type of pre textual data as well as our manually labeled sentiment data. Now our classification algorithms could detect the type of discrimination being projected as well as a Pro(Not helping the situation) of Anti(Doing things to help) types of comments.

3.3 Key Accounts Identifier

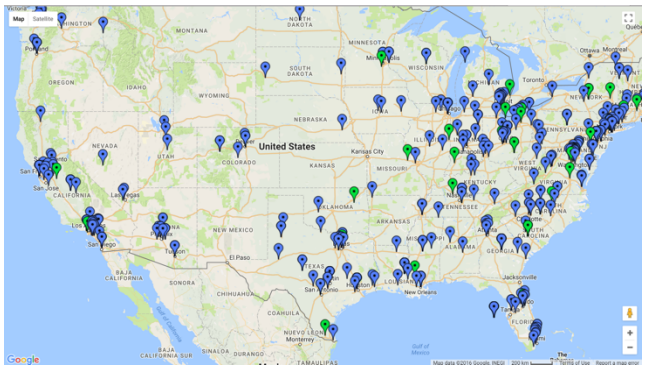
We were tasked with identifying the top-k Twitter accounts that showed more negative discriminatory attitudes compared to other accounts in that city. In a streaming approach to this solution we needed to design mechanisms to identify the top-k active either pro or anti discriminatory twitter accounts for a given city or state. For each account with data we have collected, we can look for the highest average in sentiment (Pro or Anti Discriminatory) and display this user data. Other charts created included: top cities with most tweets collected, top states, top individuals, and top cities. For our display purposes we displayed on our frontend a series of bar graphs and pie charts so that we were able to visualize this data. These charts were implemented in with the use of Chart Js a JavaScript charting tool able to display and update with real time data over and HTML page.

Top Cities with Most Tweets Collected



3.4 Online Visualization Tool

Our online visualization tool was a front-end GUI to display maps as well as relevant charts to displaying data relating to cities, states, and individuals. The first element to our frontend is the map interface. Using the python package *Flask*, we were able to setup a web framework for our python scripts to host documents and create URL routes for our files, and using the *Jinja* framework, we were able to load in saved data to be displayed in HTML form. Our scripts interface with our collected data and are then set to interface with the Google Maps API so that we can display the location a tweet came from forming heat maps around cities where discriminatory remarks are common. Markers on our map indicate the sentiment of a tweet, and hovering over them displays our classification and the tweet text. Charts were the second portion of our online visualization. The online visualization tool is also the driver for the live-streaming portion of our project it continually gets new tweets for our keywords, stores them, classifies them, and displays them on the map. When requested to do so, it also displays charted data for our collected tweets.



Online Map Display

4. Discussion

This project served as a good way to represent the widespread use of discriminatory material on twitter. Through the collection of our data, we were able to find and visualize areas where discriminatory speech is common and find users who contributed most to the situation. For twitter to appease the masses they will need to consider how to handle users whose primary objective seems to be disgruntling the masses, something our algorithm could predict would be what users are meeting this specification and doing more harm than good. Twitter could then proceed to investigate methods to limit the audience or spread of such tweets. The algorithm also has potential to help in a discriminatory context as well. If a city or state had a crisis between citizens regarding discriminatory news of events it could Identify "mediators" (those who are most active in tweeting about discriminatory material on both sides of the argument) and help to resolve conflict by identifying and convening active leasers of movements.

5. Limitations

Our model is a rather first graze attempt at identifying discriminatory data. Mostly due to the fact that our labeled set of training data is rather small. In our case, we had to collect and manually label tweets before we could even begin the rest of our project. Due to the time complexity of manually labeling tweets we were unable to feed our model with an adequate number of tweets for it to improve its classification ability. Having a larger labeled initial dataset would help immensely to better classify data. Another limitation is the use of our stopword file, we could not possibly encompass all discriminatory remarks limiting our search queries to looking for simple keywords. Discriminatory comments also have the distinct ability to change over time. To solve this problem collaboration with discriminatory experts could help to select the right words to search for, but also to help change the searches over time. Having experts label data as would help to improve the model as well. Scalability is also an issue in our algorithm, as we increase the size of our classified tweets our system become slower to respond and calculate maps and graphs. We have about 1200 tweets classified currently. As this number increases we would need to investigate optimizations in our code, and speed up our algorithm.