User Similarity Through Opinion Mining in Twitter

Chelsea Waida, Minying Lu

1.0 Abstract

We are interested to see if Twitter users' opinions are influenced by others in their social circle and whether distinct differences in tweet content can be observed among different social circles. We expect to see that users who follow a more popular user will all be more similar to each other than individuals who do not follow the same user. We also expect that a popular user will have more in common with their followers than with those who don't follow them.

2.0 Introduction

Social networks have been embedded in our everyday lives since their invention. Twitter is one of the most diverse and influential platforms for people to share their opinions on social events and has provided every user a chance to broadcast their ideas to their followers. This gives the influential power of media to individuals, for example the popular accounts on Twitter may have more followers than a local radio station has listeners. We are curious about how powerful a popular account on Twitter can be in terms of spreading ideas and how much their opinions can affect their followers' opinions. We want to find out if the choice of word, use of images and outside resources, and emotionally heavy content would affect the popularity of a tweet.

This information could be used for advertising, seeing how to phrase a certain tweet so it is more popular. It could also be used in sociology studies on people's behavior under social debates similar to Gamergate and what impact friends' opinion have on an individual. This could also be beneficial for news outlets, who would be interested in how information is circulated in social networks and further application such as using social networks as a medium for spreading ideas, news and campaigns.

We are using the 72 hours #gamergate Twitter scraped obtained from Github. Gamergate is the blanket term used for the harassment campaign, starting in August 2014 Gamergate targeted several women in the video game industry. The dataset contains user IDs and tweet IDs that used the #gamergate hashtag in a .csv file and we are using Twitter REST API to obtain the content of the tweet and information about the user.

We will do sentiment analysis and keyword extraction for all the tweets in the dataset. We will organize the keywords and polarity scores of the tweets based on the user who posted them. Then we cluster all the users depending on the keywords present in the tweet. From there, we take the most popular users in each cluster, and we find the percentage of the popular user's followers who are also in the same cluster.The experimental results suggest that the popular users do have more in common with their followers than with random users, which we detail in the next sections.

We also found that there is no strong correlation between polarity and popularity of the tweet. However, popular accounts that tweet about Gamergate have a lot of followers who tweet

about the event as well, demonstrating not necessarily a correlation of polarity, but a possible correlation for content between popular accounts and their followers.

3.0 Technique

We used sentiment analysis to look at the positive or negative sentiment of a tweet, as well as the intensity. Combined, these form a measure we refer to as polarity. Our sentiment analysis will be done with vaderSentiment, available through the nltk Sentiment package. It is a lexicon and sentiment analysis tool that is specifically designed to be used with social media expression. Since this is a more specialized sentiment analysis tool, complete a wide array of slang and emoticons, this is expected to be a much better measure than a simple, more formal lexicon.

We expect that the popularity of a tweet, a measure we define as the number of retweets plus the number of favorites a tweet has, will be highly correlated with its polarity. To test this hypothesis, we will use the statsmodel package to fit a linear regression to our data. We can then use the r-squared value, coefficient, and t and p values to determine how strongly correlated these variables are, or if there is a correlation at all.

We also performed clustering on the tweets in our dataset to group them into different opinion clusters. We apply a keyword extraction technique using the Tfidf vectorizer because it reweights the term base on frequency and recognizes a lot of the common words in English that do not contain meaningful information, keeping articles and other common but unimportant words out of our keywords.We can also add extra stopwords to filter out words in our data that we know are not valuable but are not in the default English stopword list.

After keyword extraction we have 40 feature keywords for the whole data set, so we reduce the dimension with LSD/SVD to get a normalized matrix because the KMeans clustering algorithm does not perform well with a large number of features. For clustering, we used SKLearn's KMeans function. This will group the tweets of similar content in the same cluster and users who posted the tweets in the same cluster. We expected that a single user could be in multiple different clusters because they can have multiple tweets.

4.0 Datasets and experiments.

As mentioned above, our dataset is a CSV file containing the owner user IDs and tweet IDs of about 300,000 tweets scraped over a 72-hour period during Gamergate. To collect additional information, including the text of the tweet, whether it has a URL, whether it mentions any other users, and additional information about the users themselves, we will make calls to the Twitter API. To make these calls in our code, we are using Twython, a python wrapper for the Twitter API.

First we collect the data from Twitter, collecting the owner's user ID, the number of retweets, the number of favorites, whether the tweet has a URL, the users the tweet mentions (if any) and the actual text of the tweet. This is then put into a dataframe for convenience. To do the sentiment analysis, each tweet (minus the twitter URL each tweet has) is passed into

VADER's sentiment intensity analyzer and we record the compound score (which takes into account positive and negative sentiment, and intensity). We then add these values as the Polarity value for each tweet, and add this column to the dataframe.

Next we would like to know the correlation between polarity and the popularity of the user. To do this, we will use the statsmodel package to find a linear regression on the popularity and polarity of each tweet. We will use the information provided in the summary to decide whether there is a correlation, and if so, how strong it is. This will help us decide whether stronger tweets are more popular in social networks.

For the larger part of our experiment, we will cluster all the tweets based on keyword extraction, where we will find keywords in tweets through feature extraction using Tfidf. Based on the presence or absence of our keywords in a tweet, we will be able to cluster the users using K-Means clustering. We will then take the three most popular users in each cluster and check to see what percentage of their followers who are also in the dataset end up in the same cluster, versus what percentage end up in another cluster. This would allow us to determine whether users in the same social circle really are more similar than those in different social circles, and whether this popular user may have some sort of social influence over their followers.

5.0 Results and Discussion

Our initial question was whether the polarity of a tweet would be correlated with its popularity. To answer this question, we ran a linear regression on Popularity, with the regressors being polarity, the number of mentions, whether the tweet includes a URL, and the number of retweets and favorites. Results are shown in the table below.

OLS Regression Results

Dep. Variable:	Popularity	R-squared:	1.000
Model:	OLS	Adj. R-squared:	1.000
Method:	Least Squares	F-statistic:	4.931e+31
Date:	Mon, 18 Apr 2016	Prob (F-statistic):	0.00
Time:	07:50:41	Log-Likelihood:	4.9722e+06
No. Observations:	206330	AIC:	-9.944e+06
Df Residuals:	206324	BIC:	-9.944e+06
Df Model:	5		
Covariance Type:	nonrobust		

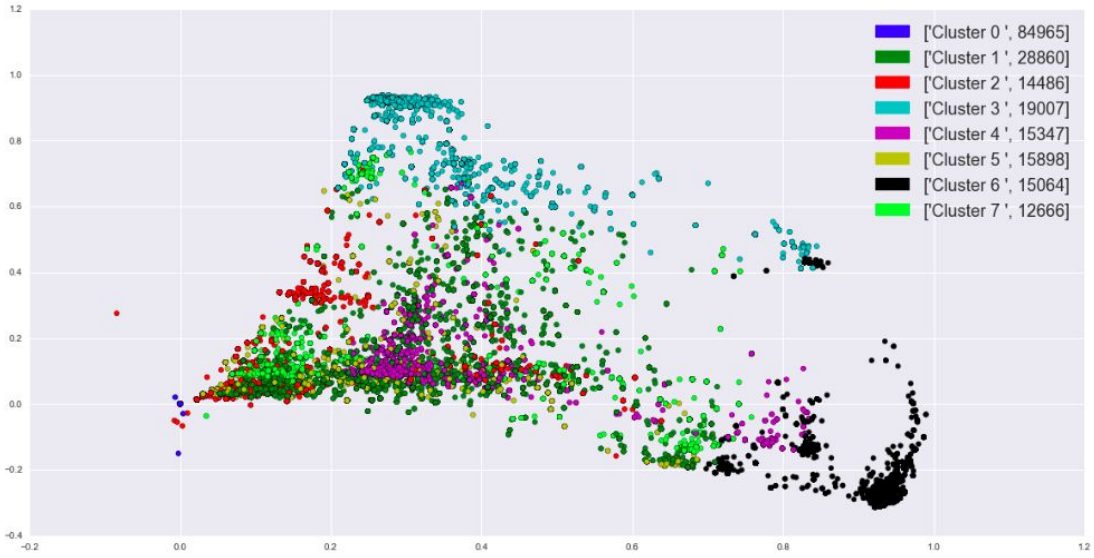
	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
Intercept	-3.624e-12	3.29e-14	-110.263	0.000	-3.69e-12	-3.56e-12
Polarity	1.963e-11	4.44e-14	441.795	0.000	1.95e-11	1.97e-11
numMentions	-3.979e-13	1.91e-14	-20.791	0.000	-4.35e-13	-3.6e-13
Has_url	1.448e-12	4.1e-14	35.336	0.000	1.37e-12	1.53e-12
Retweets	1.0000	6.44e-17	1.55e+16	0.000	1.000	1.000
Favorites	1.0000	1.58e-15	6.32e+14	0.000	1.000	1.000

Omnibus:	335.412	Durbin-Watson:	2.357
Prob(Omnibus):	0.000	Jarque-Bera (JB):	337.015
Skew:	0.099	Prob(JB):	6.58e-74
Kurtosis:	3.016	Cond. No.	794.

The regression gave an R-squared result of 1.0, which suggests a perfect fit. However, we must take into consideration that this may be a result of overfitting. The kurtosis is very close to 3, which means the distribution is close to normal. Looking at the correlation coefficients, we can see that polarity has a very low coefficient, but a high t value and a p value of .00. We predicted that tweets with higher polarity would be more popular, since they are more likely to get people

to interact with them or respond. However, even though the result is statistically significant, it is a very weak correlation, so we can conclude that there is not a substantial correlation between the popularity of a tweet and its polarity. Whether a tweet contains a URL also has a very low positive correlation, so we can again conclude there is not a substantial correlation between the popularity of a tweet and whether it contains a URL. The number of mentions in a tweet is actually negatively correlated with the popularity of a tweet. We believe this can be attributed to the users who use mentions as a one-on-one conversation with other users. These tweets always contain at least one mention, but do not get many retweets or likes since they are only meant for one user. Naturally retweets and favorites have a perfect 1.0 correlation coefficient with popularity since popularity is computed based on these two values.

After clustering tweets based on keywords found by Tfidf keyword extraction, we are able to show the following plot.



Here we see a very dense area of points, and many clusters overlapping. This is partially due to the nature of Twitter, where users can retweet each other and we have the same text being put out by a number of unique users. However, it also demonstrates that many users are discussing the same topics related to the issue. The outliers tend to be tweets that have fewer keywords in common with their cluster, and because of this are almost always original tweets instead of retweets. Clusters 3 and 6, which are both further from the dense area of clusters, are not as similar because they are not centered around the same retweets the other clusters are, therefore making them more distanced than many of the other clusters.

When we find the owners of the three most popular tweets in each cluster, we find overall that there are high percentages of the user's followers who are also in their cluster, the majority having at least 60% of their followers in the same cluster they are in. Since we have 8 different clusters, this is much higher than chance, and shows significant similarity between users and their followers. This being said, we did have a few values where this percentage was much lower than expected. One owner of a popular tweet was actually a user who only had 73

followers and had simply retweeted a tweet that had already received a lot of attention, and 0% of his followers were in his cluster. However, this was simply because none of his followers had tweeted during this 72 hour scrape, and were therefore not in our dataset. To try to limit this type of problem, our ratio measure computed the number of a user's follower's in the same cluster as the user, divided by the total number of the user's followers in our dataset. Also the users in this cluster are mainly discussing the victim, who is the owner of the original popular tweet.

Another very interesting result we found is that a some of the users in our dataset tweeted hundreds and thousands of times during the 72-hour period in which this data was scraped. For example, the most "talkative" user in our dataset tweeted about 2,000 times. He responds to other users' tweet by mentioning them in a new tweet, and as a matter of fact he created his Twitter account in order to join the Gamergate discussion on Twitter. Since he tweeted so much, he has one of the highest scores in our dataset. This make sense because when he tweeted and retweeted, he influenced a lot of people, including his followers and people who he responded to.

6.0 Conclusion

Our project had the intention of determining whether Twitter users' opinions are influenced by others in their social circle and whether distinct differences in tweet content can be observed among different social circles.The fact that our ratio measure was generally high demonstrates that popular users often have a large percentage of their followers in the same cluster as they are in, helping us to conclude that there is greater similarity between a user and their followers than between random users. Therefore, users can be influenced by others in their social circles.

We can also see that there are distinct differences in tweet content among different social circles, however these differences are not as clearcut as we expected. Many of the clusters we found overlap, demonstrating a degree of similarity within the different clusters. This is likely due to the way Twitter is set up, since the concept of retweets makes it possible for many different users to post the exact same tweet, causing the clustering to be much less clean that it may have been if all tweets were distinct. Still, using keyword extraction we are able to cluster tweets by topic, and it is clear by the high percentage of users' followers being in the same cluster as them, that users who tweet about the same things tend to follow each other. Therefore, there is a high degree of similarity in the content of tweets in the social circles we observed.

In the future, it would be interesting to look more closely at the composition of social circles. We have determined that popular users tweet about similar things that their followers tweet about, but it would also be interesting to see how their followers are related to each other. Here we defined social circles as a popular user and their followers. We would expect, however, that each user is part of many different social circles. In future work, we would be interested in

analyzing how these multiple social circles are connected, and whether our findings would hold for more intricate social networks.