CSCE 578 Final Project

Noemi Glaeser & Nick Quan
24 April 2019

1 Overview

We conducted several analyses on the corpus of our shared friend group chat, hereafter referred to as "Spyral", which we have been part of over all four years of college. They are described in detail in Sections 3 through 6. Our code can be found on GitHub at `https://github.com/nglaeser/csce578final`.

2 Methods

Specific methodologies are discussed in the respective sections below. In each case, however, we use the GroupMe API and a GroupMe fetch script found at `https://github.com/cdzombak/groupme-tools` to pull a full transcript of the messages in Spyral. This data comes in the form of a JSON file and was easily accessed using Python's `json` package.

Throughout this paper, asterisks are used to denote bots.

3 Clustering

This portion is an improved version of Noemi's Assignment 02, which uses `tf-idf` values to group people based on what they talk about. The stopword list was incrementally expanded to include other words which showed up in the output but were not elightening. Though initially based on the stopword list provided on the Moodle site, the stopwords have expanded to include "www", "http", "https", "com", "lol", "yeah", "like", "m" (for "am"), "also", "okay", "ll" (for "will"), "xd" (the laughing face), "ve" (for "have"), and "d" (for "had"). This addressed the URLs and other uninteresting "words" that came up in people's top `idf` values.

We also improved anonymity so that we would be able to publish the top `idf` terms and values for each user. Previously, users' names would come up in these lists, but they have been replaced with the pseudonyms utilized in all parts of the assignment.

Another improvement was the removal of emojis and other non-ASCII text. We felt that comparing text to text was more illustrative, though in the future, an analysis of each user's characteristic emojis could be interesting.

In the future, `tf-idf` values could also be weighted by how widespread the words are in the particular user's corpus; this would mitigate instances in which, for example, a single word appears at the top of one individual's `tf-idf` list simply because it is fairly obscure and was used several times in one single conversation.

3.1 Methods

1. Obtain a transcript for the GroupMe group to be analyzed as described in Section 2.
2. Read in the stopword list and add any additional words
3. Read in the transcript. For each message:
   - Skip system and calendar messages
   - Clean out punctuation
   - Remove stopwords
   - Remove non-ASCII characters
   - Remove any user's names that come up in the message and replace them with the user's alias (for anonymity)
   - In the dictionary of each user's word counts, increment the counts corresponding to the words obtained
4. Obtain a `tf-idf` value for every word in the user word count dictionary
5. Sort each user's words by `tf-idf` and print the top 10
6. Populate a matrix with $cos(\alpha)$ values to approximate user-user similarity
7. Print the top 10 similar users
8. Do singular-value decomposition (SVD) on the vectors of users' top `tf-idf`s
9. Reduce the dimensions of each vector to 2 dimensions and plot each point

3.2 Results

Because of the elimination of more stopwords and emojis, the list of top ten similar users changed slightly from the results in Noemi's Assignment 02. The new list is shown in Table 1.

The plots obtained in step 9 are shown in Figures 1 and 2. Each point is labeled with a number which corresponds to a user according to the key in Table 2.

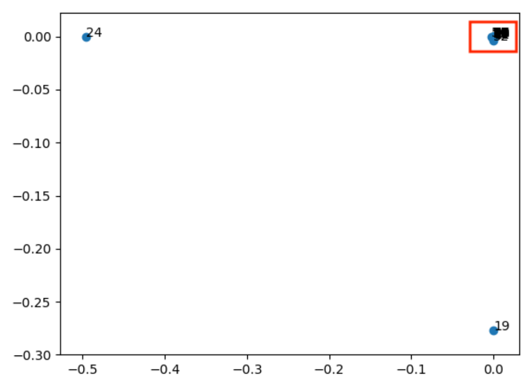
3.3 Analysis

There is no quantitative way to truly analyze these results and confirm them. Overall, however, the graphs seem to do a pretty good job of grouping members who are more active in the group than those who only contributed a couple of times.

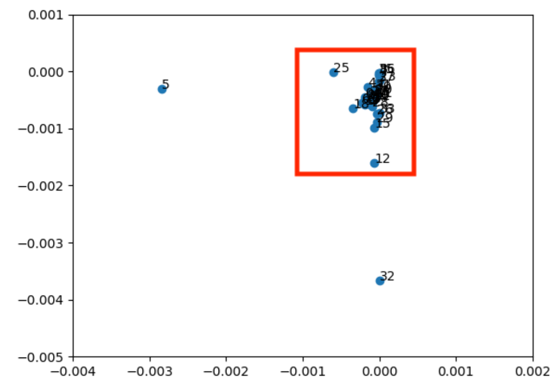
There are several clusters that can be seen. First, 19 and 24* are far outliers, and neither of them spoke much in the group. Next, 5 and 32 stick out – again, both are users who

User pair	$\cos\alpha$
0, 9	0.8858758947976005
13, 16	0.8627483384717963
2, 27	0.8362452991879455
13, 27	0.8357123185575374
16, 21	0.8045717280102884
16, 27	0.8017144592623564
3, 13	0.7937801422111465
21, 27	0.7902834211530108
6, 13	0.7873164005896532
15, 16	0.7819017671529359

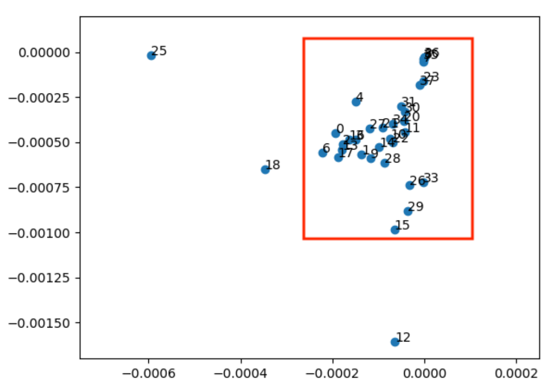
Table 1: Similar user message corpora based on *tf-idf* clustering.



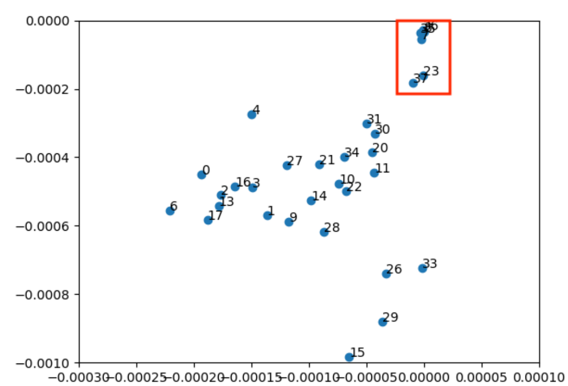
(a) Full plot of user clustering. The red box is shown in more detail in Figure 1b.



(b) Detail view of Figure 1a. The red box is shown in more detail in Figure 1c.



(c) Detail view of Figure 1b. The red box is shown in more detail in Figure 1d.



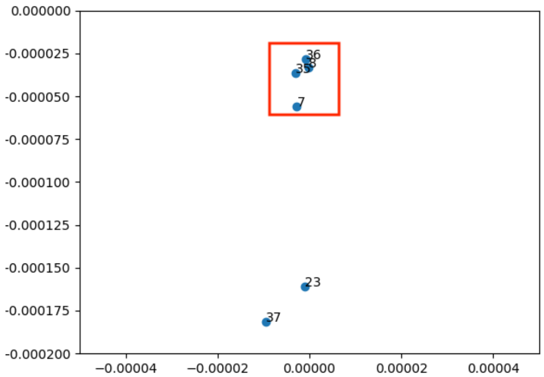
(d) Detail view of Figure 1c. The red box is shown in more detail in Figure 2a.

Figure 1: Plot of the characteristic vector of each user (reduced to two dimensions). Numbers correspond to users as described in Table 2.

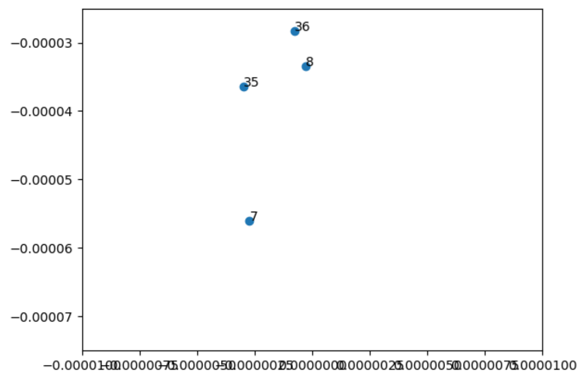
Number	User
0	Nathan
1	Trey
2	Jim
3	Lisa
4	Nina
5	Chris
6	Jacob
7	Anna
8	Kylie
9	Ethan
10	Sally
11	Sarah
12	Simon
13	Nancy
14	Peter
15	Walter
16	Jill
17	Alex
18	Aaron
19	Mike
20	John
21	Annie
22	Michael
23	Alan*
24	Tom*
25	Bob
26	Jason
27	Nicole
28	Thomas
29	Liza
30	Isaac
31	Jack
32	Patrick
33	Tim
34	Claire
35	Mitch
36	Zo*
37	Natalie

Table 2: Key for clustering plots in Figures 1 and 2. Asterisks indicate bots.

only made rare appearances to ask a couple questions near the start of Spyral's history. The same goes for the next set, 12, 18, and 25.



(a) Detail view of Figure 1d. The red box is shown in more detail in Figure 2b.



(b) Detail view of Figure 2a.

Figure 2: Plot of the top right cluster in Figure 1d.

We can see from Table 1 that despite the dimensionality reduction, the plots do a fairly good job of keeping user pairs deemed similar by the $\cos\alpha$ method near each other. With the excepton of (0,9) and (15,16), the pairs listed are fairly close to each other in Figure 1d.

The large cluster in the red box in Figure 1c contains all the active members of the chat. They are mostly clustered in one big blob together; besides this major grouping, two others stick out: 15, 26, 29, and 33; and 7, 8, 23*, 35, 36*, 37 (shown in more detail in Figure 2.

The top `tf-idf` terms for the first group are shown in Table 3 and in Table 4. Just by looking at these lists, it is unclear why these two groups should tend to cluster together. As such, they should be taken with a grain of salt, since we have seen that nearby points don't necessarily have high $\cos\alpha$ points and vice versa: not all similar points appear near each other on the projection used in these plots.

(a) Walter (15)

Word	`tf-idf`
guys	0.00479
anyone	0.00462
honeycombe	0.00400
jesus	0.00379
could	0.00368
sure	0.00351
lisa	0.00351
trey	0.00344
go	0.00323
going	0.00320

(b) Jason (26)

Word	`tf-idf`
otw	0.00779
valafar	0.00680
capa	0.00584
anyone	0.00476
lon	0.00473
comtab	0.00445
bookmark	0.00389
blobby	0.00388
didn	0.00320
dinner	0.00291

(c) Liza (29)

Word	`tf-idf`
douses	0.01219
gibbes	0.01158
egg	0.01038
gala	0.00986
going	0.00953
anyone	0.00801
eczema	0.00754
banquet	0.00724
hotbar	0.00609
heeeyyy	0.00609

(d) Tim (33)

Word	`tf-idf`
prop	0.03497
wally	0.03136
habichi	0.03136
hijacking	0.02538
dynamics	0.02189
ummmmm	0.02189
jackass	0.02189
hottest	0.01941
lisa	0.01748
din	0.01722

Table 3: Top `tf-idf` words of users in the first cluster discussed.

(a) Anna (7)

Word	`tf-idf`
rejoined	0.04664
beta	0.04338
memeage	0.03775
junipero	0.03775
rejoin	0.03255
theta	0.03255
sup	0.02886
delta	0.02886
haunts	0.02886
gf	0.02600

(b) Kylie (8)

Word	`tf-idf`
bloodmoon	0.13991
observatory	0.11325
slut	0.09765
cough	0.09536
livestream	0.07801
lunar	0.06506
nasa	0.05135
stream	0.04433
total	0.04126
site	0.03840

(c) Mitch (35)

Word	`tf-idf`
engaged	0.07853
lizstletoe	0.06310
holly	0.06310
yo	0.04395
awkward	0.04395
wasn	0.03215
guess	0.03215
wrong	0.02703
cool	0.02463
even	0.02463

(d) Zo (36)

Word	`tf-idf`
zo	0.10313
trivai	0.09981
erybody	0.06165
secs	0.04991
msg	0.04303
privacy	0.04303
smartest	0.03438
agreement	0.03438
sooo	0.02867
display	0.02641

Table 4: Top `tf-idf` words of users in the second cluster discussed.

4 Text Complexity

We initially also planned to run a continuation of Noemi's Assignment 03 (using the Stanford POS tagger to determine average sentence complexity with element tree depth).

Unfortunately, the other portions of this assignment took up more time than anticipated. Thus, we were unable to focus on extending and improving this work. This included confirming the accuracy of the element tree depths obtained with the parser and possibly applying additional complexity measures used in the *Washington Post* article discussed in class which analyzed political language.

5 Sentiment Analysis

5.1 Methods

1. The JSON data was read in for the specified day range depending on the sentiment code ran (either 8/16/2015 to 4/13/2019, 1/1/2018 to 12/28/2018, or holidays in the year 2018).

2. `NLTK VADER` and `TextBlob` sentiment scores were calculated for each area of interest. Scores were defined as the mean score of each day when looking at a day of the week; "positivity" (polarity) scores when looking at day-to-day mood changes, weekday mood, message count changes, holiday mood changes, exam mood changes, and Fourier transform cycles. Scores were defined as the sum of all scores for each day for a Fourier transform to look for cycles as well.

3. The scores were graphed using `matplotlib`.

5.2 Results and Analysis

We were very interested in looking for patterns and shifts in mood over time in the Spyral chat. In overall society, Mondays are generally considered a low point in the week, as evidenced by Garfield comics and day-to-day banter. We wondered if people's moods decrease around Monday and increase near the weekend, decrease around exam times with stressful messages, and increase around holidays with celebration. From our text analysis, we cannot conclude that an individual's mood is completely revealed or indicated in their messages, but they may still suggest interesting trends nonetheless. Due to the vast amount of different sentiment analyses possible as well as time constraints, we decided to focus on seven different ideas.

All seven of the ideas started by calculating sentence polarity scores over different time periods. `NLTK VADER` and `TextBlob` both accept emojis, making it even easier to work with the message data.

Calculating the sentiment scores for each day was fairly quick (around 5 minutes to calculate the scores for 4 years of chat data). 7 Python scripts were written to analyze different trends.

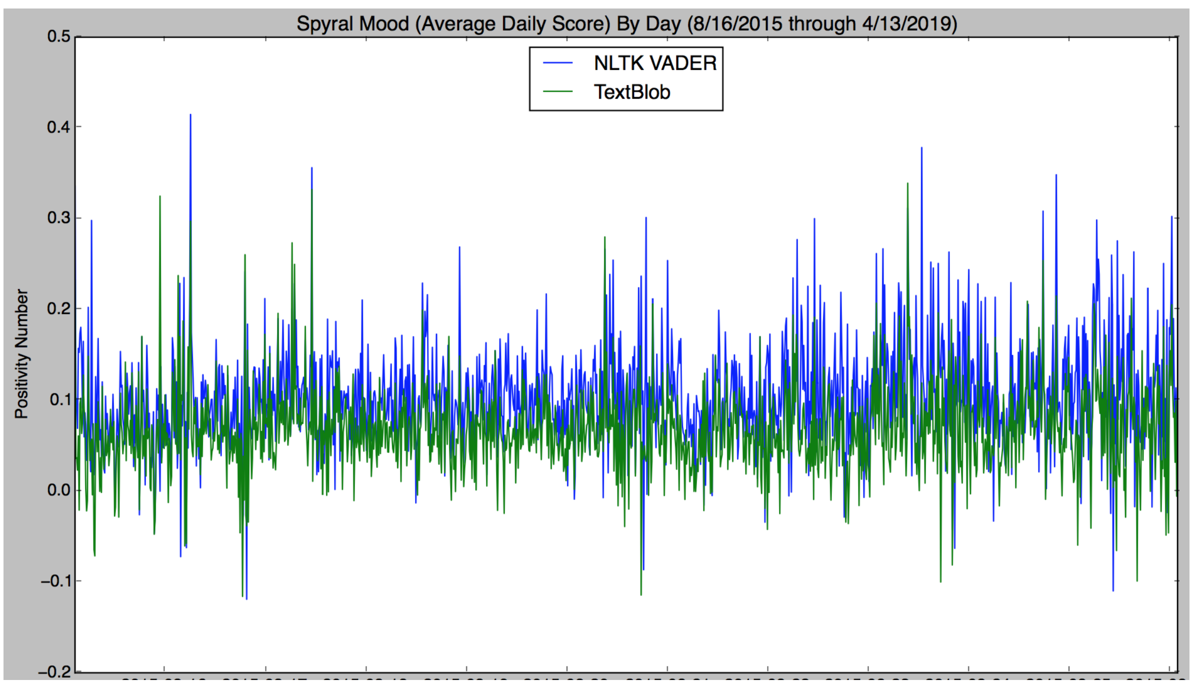


Figure 3: A time series plot of average daily mood scores from 2015 through 2019

As seen in Figure 3, the polarity scores (also referred to as "positivity number") are calculated with `NLTK VADER` and `TextBlob` for each message in each day and averaged together to get the average score for the day. The plot indicates that `NLTK VADER` tended to score messages higher than `TextBlob`, since the blue line is generally above the green line throughout the plot. Additionally, we can see that almost all the days have a score above 0.0, suggesting that on the whole, the members of the chat share more positivity with each other than negativity. Very few days fall below 0.0, with some days spiking below -0.1, and some days spiking over 0.3. To put this in perspective, a very positive message will be close to 1, and a very negative message will be close to -1.

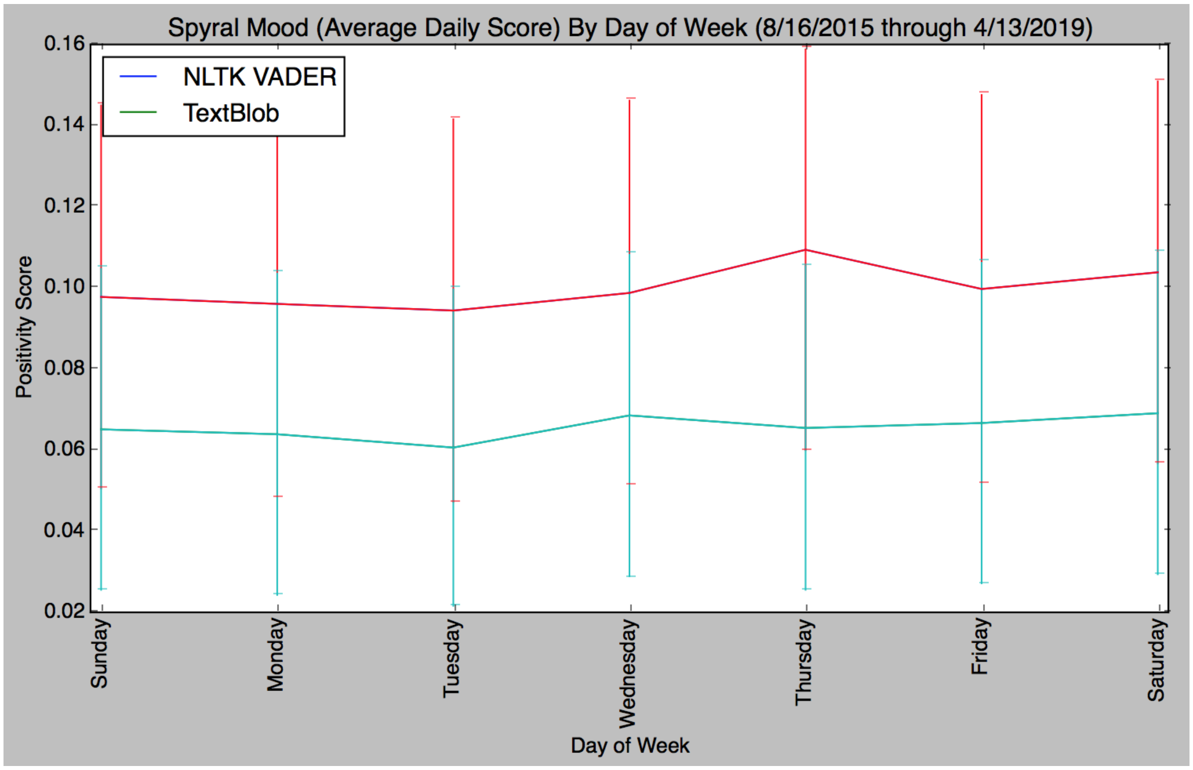


Figure 4: A plot of average mood on each of the 7 days of the week from 2015 through 2019

Figure 4 shows the average positivity score of each day of the week using the chat messages from 2015 through 2019. 95% error bars were calculated for each day of the week. Again, `NLTK VADER` tends to show a higher score for positivity every day of the week compared to `TextBlob`. An interesting difference appears on Thursday, where `NLTK VADER` scores rise in positivity, whereas `TextBlob` scores fall. Future analysis could be conducted to determine what messages led to this difference.

The error bars suggest there is not a significant difference between any of the days. We can still propose some non-significant trends. Based on the lines themselves, we were surprised to see that Monday did not have the lowest positivity score, but this fell on Tuesday instead. It is possible that people's moods have been worn down by Tuesday due to the events of Monday, and so they are subsequently lower on Tuesday.

Thursday is also a surprising day. We expected the positivity score to rise on Thursday because the group chat has had a trend for several years now called "Thankful Thursday", in which people share things they are thankful for on this day of the week. These messages would definitely be positive, but perhaps on the whole it does not make a great difference because of the number of messages sent each day. Also surprising was that the positivity score on Friday was not higher than the other days. We had expected Friday and perhaps the weekend days to have more positive messages because no classes take place on the weekends, but there doesn't seem to be a noticeable difference.

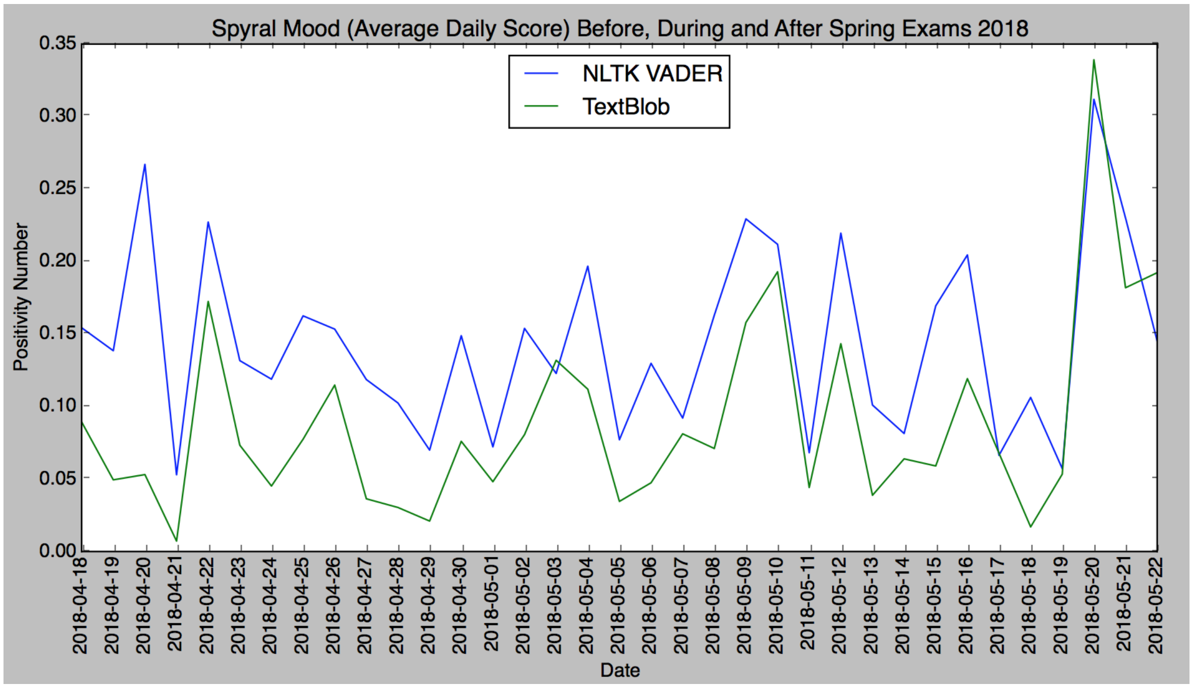


Figure 5: A plot that shows the daily average mood around exam time in Spring 2018

Figure 5 shows the analysis about two weeks before and after exam time in 2018 (approximately May 2-8). We had expected the positivity scores to be lower around the exam dates, but they are still positive and hovering around the average (approximately 0.1 for `NLTK VADER` and 0.05 for `TextBlob`). It is difficult to speculate given the plot, but May 9th may have been higher than the previous days since it marked the end of exams.

Additionally, There are positivity spikes seen on April 20 and May 20, though the reason is unclear. It is possible that the one observed on April 20 is due to jokes about that date. To determine the sources of these spikes in positivity, as well as others, further analysis should be conducted to see what messages and events led to these patterns.

In Figure 6, 2 indicates New Year's, 5 is Independence day, 8 is Thanksgiving, and 11 is Christmas. The numbers to the left and right of these are the days before and after each of the listed holidays, respectively.

The day after New Year's had the highest positivity score out of all the days around the four holidays. Further analysis needs to be conducted to determine what exactly was said the day after New Year's that made it more positive than New Year's Day itself. Perhaps people were celebrating, especially with family, during January 1st and did not have time to discuss until the day after. This hypothesis may not be very tenable, however, as other holidays tended to have the highest positivity score on the holiday itself, not before or after (at least for the `NLTK VADER` data). `TextBlob` appeared to differ a lot from `NLTK VADER` during Thanksgiving and Christmas. It showed a higher positivity score on Thanksgiving day whereas `NLTK VADER` computed it was more positive the day before Thanksgiving. On Christmas, `TextBlob` didn't show a spike in positivity whereas NTLK VADER did. These differences are likely due to different interpretations of certain types of words, such as words of gratitude or perhaps holiday-themed words.

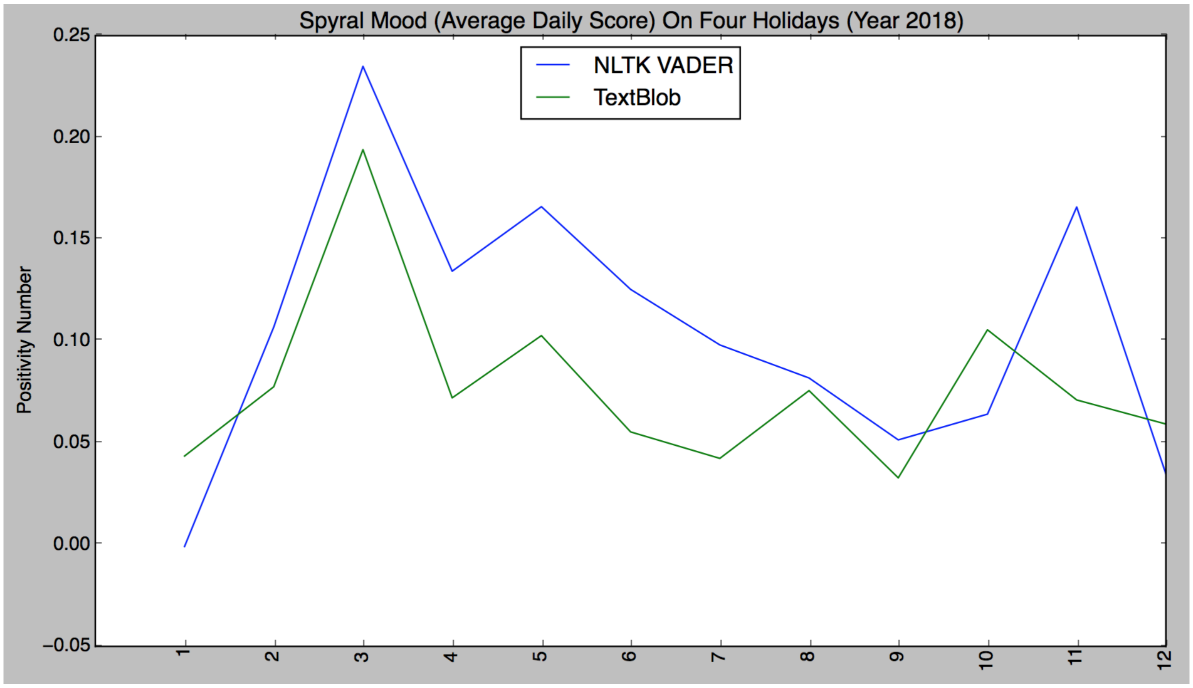


Figure 6: A plot of average mood scores around New Year's, Independence Day, Thanksgiving and Christmas in 2018

In Figure 7, the x-axis refers to the number of cycles per 365-day period, and the y-axis refers to the power. (As a disclaimer, we ar not very familiar with Fourier Transforms or Power Spectrum Density Plots, though we believe we performed the correct calculations on the data and can make a valid analysis.)

To create the plot in Figure 7, first a Fast Fourier Transform was run on the average daily positivity scores for `NLTK VADER` (TextBlob was not analyzed). Next, the calculations were performed to convert this into PSD plot. As seen in the above graph, there do not appear to be any significant cycles that stand out.

Figure 8 shows the PSD plot with all the positivity scores in each day summed together rather than averaged. We thought this may show some visible patterns, since from personal experience, there seemed to be less messages on weekends than weekdays. Having less messages on weekends could lead to less positivity to be summed together on weekends.

This PSD plot looks much different than the previous one, where scores are averaged. In this summed score plot, there is a big spike near 1, which makes sense because it reflects each year period being different from another. There is also a spike at 52, which is perhaps more significant. 52 means that that there is a cycle 52 times a year, or in other words, once per week. This cycle could suggest that something interesting is happening at the week time period level. It is important to remember that this cycle did not show up in the average score PSD plot. The difference could be explained by the number of messages sent each day, which would impact the sum of the positivity scores but not the average positivity scores.

Figure 9 shows the number of messages sent on each day of the week over the 4-year period. Sunday and Saturday show the lowest number of messages, as expected. From personal experience, we have noticed that people tend to send less messages on the weekend.

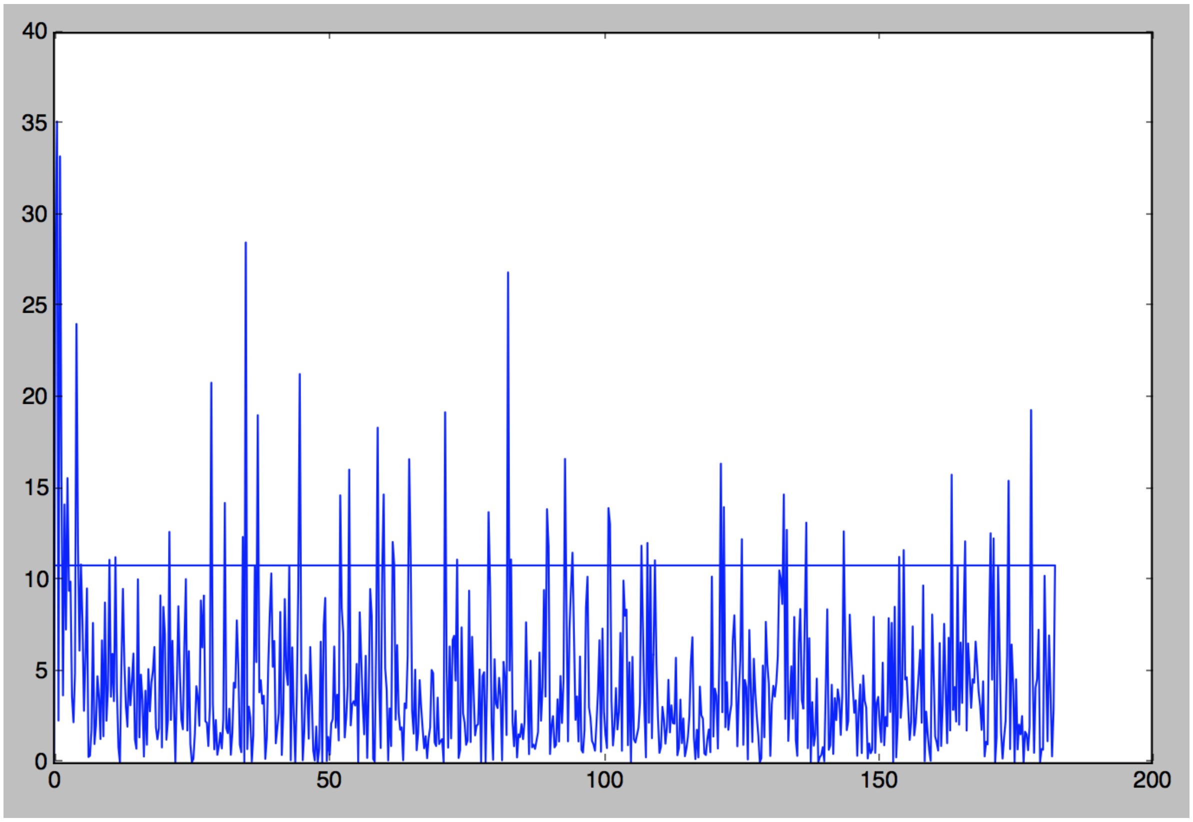


Figure 7: A power spectral density plot of the average mood scores from 2015 through 2019

This data provides empirical evidence that this is the case. To quantify this difference: the middle days of the week have around 25% more messages than the weekend days. This plot could explain that in the summed positivity score PSD plot, a cycle shows up simply because there is less to sum on Sunday and Saturday.

We hypothesize that people send more messages during the week because they are making comments about school-related topics. On weekends, people in the chat are not attending class, so there may be less information pertinent to other members on these days. Another possibility is that people are relaxing, not having as much to comment about and consequently are not posting as many messages.

6 Topic Modelling

6.1 Methods

1. JSON data from the GroupMe was obtained for 1/1/2018-12/28/2018.

2. The `NLTK RegexpTokenzier` was used to tokenize the documents, excluding stopwords. A dictionary of words and a document-term matrix were created in preparation for LSI using `gensim`.

3. `gensim`'s `tf-idf` function was used to convert the document-term matrix to a `tf-idf`

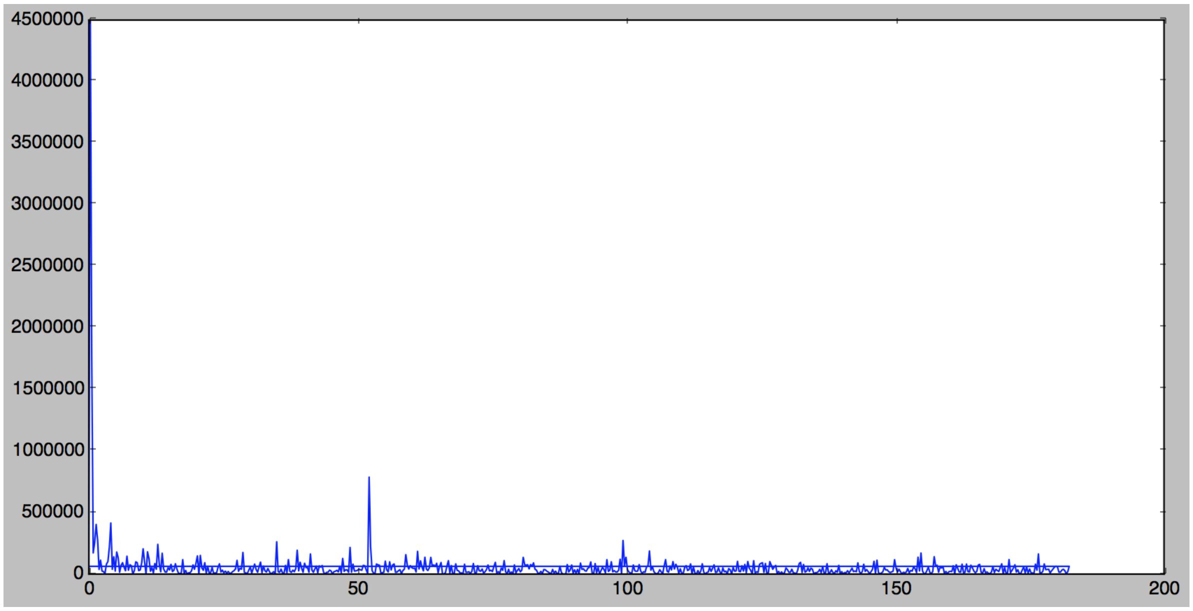


Figure 8: A power spectral density plot of the mood scores of each day summed from 2015 through 2019

document-term matrix.

4. An LSI model with 5 topics and 5 descriptor words for each topic was conducted on each month of 2018.

6.2 Results

For this part of the project, the aim was to run topic modeling on the group chat data to see if topics could be constructed. This part follows the methods in Nick's Assignmet 03 fairly closely, besides adaptations made to have run on group chat messages instead and attempted filtering to obtain more comprehensible topics. `gensim`'s `tf-idf` was used to form the corpus. `gensim`'s LSI was run on each month of chat data from 2018 (only using 28 days in each month for coding simplicity) to produce 5 topics per month with 5 descriptor words for each topic. The number of topics (5) was chosen by running coherence score calculations like in Nick's Assignment 03 and by manually analyzing different coherence numbers. Overall, 5 was the most coherent number between the coherence scores and our interpretation.

The output for the first 6 months of 2018 is shown in Figure 10. As seen in any of the months and topics, there doesn't seem to be much correlation between different words, for the most part. In Month 1, "Happy" suggests someone saying happy birthday, whereas "Russell" suggests that people want to eat at the Russell House. For Month 2, "China" shows up multiple times, which makes sense because a member of the group chat was in China at the time. "Hibachi" also shows up this month because people like to eat there. For many of the months the words "Thankful" and "Thursday" appear, which suggests the topic of Thankful Thursday, an aforementioned trend in Spyral. "UVA", "UMBC" and "Duke" show up in Month 3, which is March. This makes sense because these teams competed in

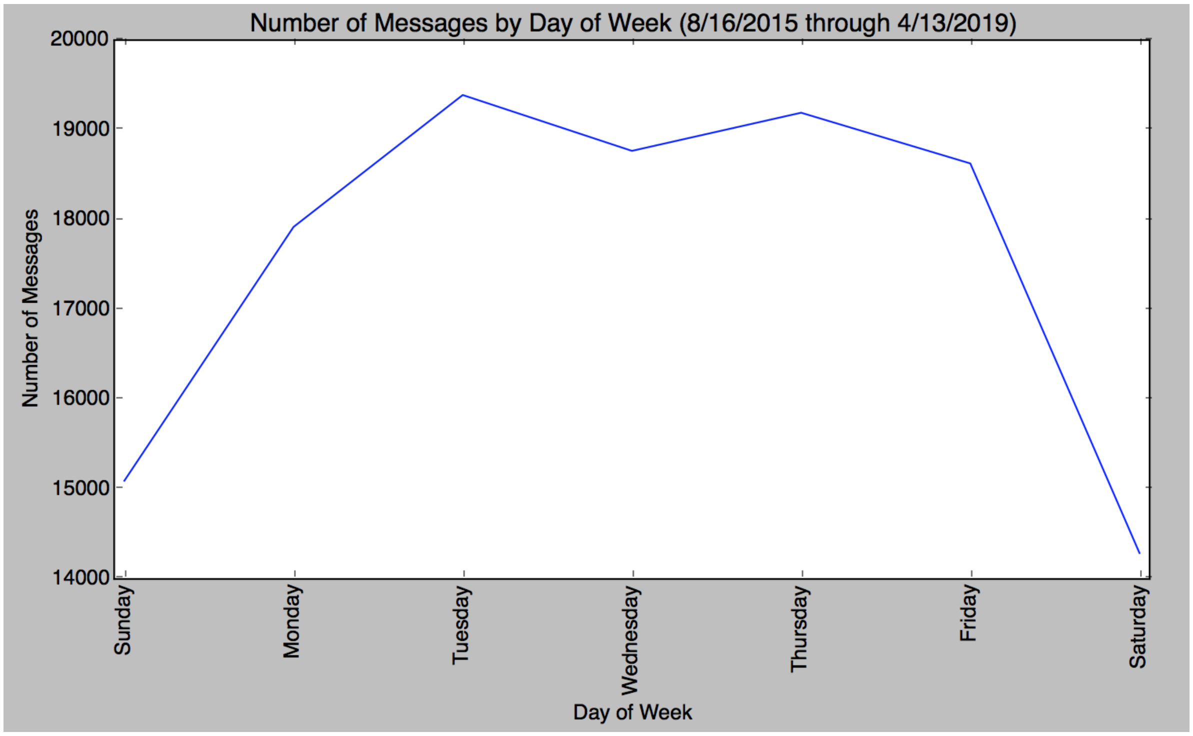


Figure 9: A plot of the number of messages sent on each day of the week

March Madness. "Happy" and "Birthday" show up more times throughout, "Oslo" shows up in Month 5, which was when one of the chat members came back from an Oslo, Norway, where he studied abroad.

Overall, we can see that the technique used on the chat data was not super effective. The first reason for this is because there may be many insignificant conversational words that need to be removed as stopwords. Along with this, it could be because there are less long-term topic trends in the group chat. Looking for topics over each day could come up with much more comprehensible subject matter than trying to find trends over whole month periods. For instance, Christmas would be more likely to show up in the week of Christmas than the first week of December.

One hurdle to overcome in the analysis was that many of the chat users' nicknames were mentioned throughout the transcript and had to be removed. Luckily, GroupMe records every nickname each user has taken, which simplified this task.

Another problem was filtering out unimportant words like "Men", "As", "On" and "So". This could be fixed in the future by converting all the words to lowercase so that capitalized stopwords could be omitted from the transcript. "She" and "He" showed up because we didn't convert them to lowercase, which would be an easy fix as they are in the stopword list in lowercase form. The filtering could also be improved by manually adding words to the stopwords list. As I previously alluded to, there appear to be many conversational stopwords that were not in the list we have been using this semester. Some of these words include "Sure" and "Yeah".

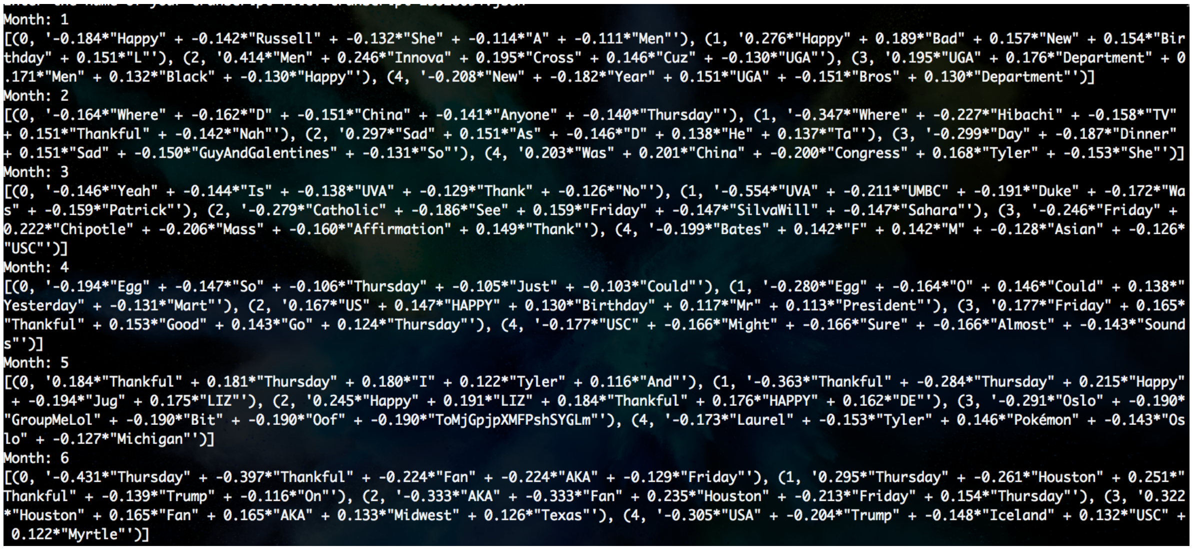


Figure 10: Output for the first 6 months of 2018

7 Conclusion

On the whole, these analyses were an interesting study of our friend group's dynamics and character. Clustering users was far from perfect, and additional strides can be made to only consider words which truly make sense as representatives of a user's style and interests. The $\cos\alpha$ similarities and the actual plots obtained through SVD dimensionality reduction were close enough but did not completely agree. In particular, it would be interesting to find a way to better represent these similarities visually, whether in a three-dimensional plot instead or some completely unrelated manner.

When analyzing sentiment and topics, future work could study each member of the chat individually. The data may be too noisy when averaging 20 peoples' messages together, but perhaps looking at individuals over time would show interesting trends in mood shifts and topics. It could be the case that certain individuals' moods are highly affected by the day of the week, exam periods and holidays, while others express themselves differently in these periods. The same idea could be applied to topics. Some individuals may like to talk about specific topics, which may be elucidated through an analysis of individuals' messages.

Along with observing individuals, a closer look at smaller time periods, for example the course of one day, may allow for a greater understanding of common sentiment-altering words as well as topics. This current analysis looked more generally as a first time look, so analyzing smaller timescales would be the next practical step.

Along these lines, high and low spikes in sentiment could be analyzed to determine what causes them to occur. Additionally, the different Python libraries for sentiment analysis used could be compared to determine what causes the significant differences in sentiment score magnitude. Both NLTK VADER and `TextBlob` have sentiment classifiers that classify sentences from -1 to +1, but `NLTK VADER` has much higher polarity scores (referred to as positivity throughout this project). The difference in scores could be analyzed to see if there is constant factor of difference so that the score magnitudes can be normalized. This discovery could allow sentiment calculated by different systems to be compared numerically.

Despite the differences in the classifiers, they still both made it possible to analyze the sentiment of group chat messages, so rough analyses can currently be conducted. Creation of a sentiment analysis library from scratch could potentially increase accuracy of sentiment analyses even further.