

GroupMe and Discord Content Analysis and Comparison Using APIs

Kevin Ye and Taylor Richardson

Abstract— This project will look at the speech patterns and habits between users of two of the most common chat applications at the University of Tennessee, Discord and GroupMe. Data for the project is gathered via usage of the respective APIs for both applications. By employing these APIs, one can examine not only the kind of communication that occurs but also when it occurs.

We explore two chats within a larger group and see how the chats function in terms of discourse. Each application has professional and casual sides, but the goal is to see where the line is drawn between these two diverging purposes.

I. INTRODUCTION

GroupMe and Discord are two common social media applications that students at the University of Tennessee use to coordinate with each other. Both applications differ in their own ways, such as Discord having audio and chat channels while GroupMe emphasizes easy-to-create chat groups. GroupMe chats are typically created on a class by class basis, with clubs and organizations also having their own chats. Discord chats are usually based around an umbrella organization like the EECS department or the Esports Club, with separate chats being contained within for certain classes or groups. The goals of this project are to analyze the way students interact on each of these platforms and see how posted content differs through a variety of text analytics.

II. BACKGROUND

GroupMe and Discord are both social media applications that are popular among students and organizations alike. Some organizations, like the Esports club, moved club communications from GroupMe to Discord.

A. GroupMe

GroupMe was conceived in 2010 at a hackathon in New York and released later the same year[1]. The application evolved from a simple way to create private SMS groups to full mobile app with a complementary web application and client. It has become widespread among colleges for the ability to easily create group chats for both serious discourse, such as in an officer GroupMe, or for entertainment, such as “meme” GroupMe chats. These are chats where users post funny pictures and discuss topics like TV shows, cartoons, and games for fun.

It is notable that GroupMe has built in features to facilitate the latter type of conversation; any uploaded picture can be easily “memed”, or captioned in the style of popular internet memes. Pictures can also be recaptured and easily uploaded.

B. Discord

Discord was developed by Hammer & Chisel, a game development studio founded in 2012[2]. Utilizing funding obtained through YouWeb’s 9+ incubator, Discord was officially released in May 2015[3]. Discord functions similar to the way an IRC chat server does. While originally intended for gamers, its consistent stream of updates and effective branding have helped it branch out into multiple communities. The application’s ease of use allows users to organize large communities by creating separate chat rooms for whatever niche topic a group has an interest in. Discord has an advantage over GroupMe in that it allows voice chat, with video chat coming in the near future. The Voice-over-IP option, or VoIP, has allowed Discord to increase its user base by appealing to users of Skype.

C. Motivation

The Esports club at UTK, a competitive gaming club, was founded in 2016 after evolving from the League of Legends club, an organization devoted to playing and competing in the game of League of Legends. Originally, the club had two GroupMe chats, one for serious communication, and one for casual chats.

Since then, the club has moved to Discord for most of its communication but the GroupMe chats still exist and are still active. With the goal of looking to improve club communication, both authors, who are officers in the club, will analyze posted content in each application. Obviously, collection will only occur after member permission has been acquired.

They will look to establish metrics to qualitatively and quantitatively define the text and images posted in chat for both GroupMe and Discord. Those metrics will then be used to compare the two platforms.

III. METHODS

The proposed analysis will use the following APIs to complete the project. The denoted milestones will mark project progression and represent short-term goals.

A. GroupMe API

GroupMe's API is a simple REST API. A bot or application is manually associated with a specific group, given an identifying token, and is added as a user to a given group. From there, the bot or application can post and receive messages via JSON request envelopes [4].

Beyond simple group interaction, a bot or application can create groups, message people, and view leaderboard data. Leaderboard data tracks popular messages in each group [5].

While the idea of using a bot to collect the message data was introduced as a possibility in the proposal, it ended up being easier to simply use the application API to query the GroupMe chats.

This process entailed writing a python script using the requests library to make GET requests to the GroupMe API using a developer authentication token. The resulting JSON objects returned by the requests were parsed and stored in a Mongo database accessed via the pymongo library.

The GET request, while it only can return a finite number of posts, had a parameter specifying an upper bound on message ids so it would fetch messages that came before a given message id. Using this in a loop, it was possible to grab messages indefinitely back to the creation of a given groupme.

The database queries were then processed via matplotlib.pyplot to generate the various histograms and graphs seen.

B. Discord API

Discord's API is based around two core layers, a HTTPS/REST API for general operations, and a persistent secure websocket-based connection for sending and subscribing to real-time events[6]. While the API lets us capture the unique user ID attached to every message, we'll be ignoring it to protect the privacy of users. We can also track the differences between admins and normal users, so we can observe how users with special privileges behave differently compared to normal users.

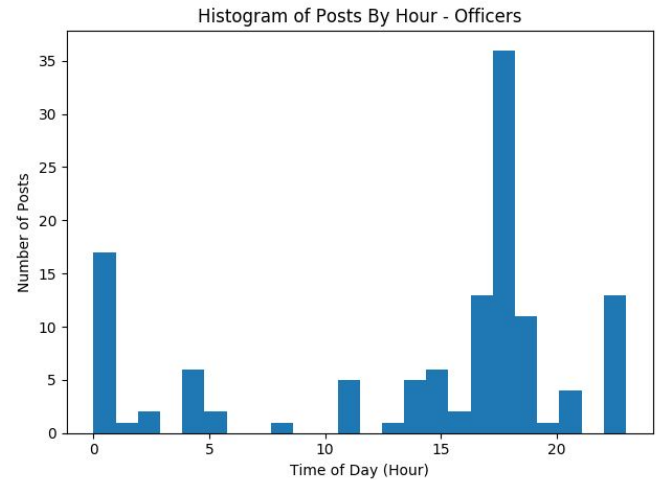
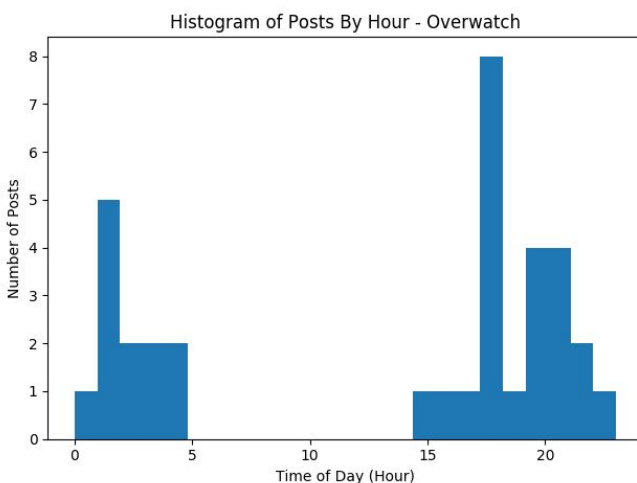
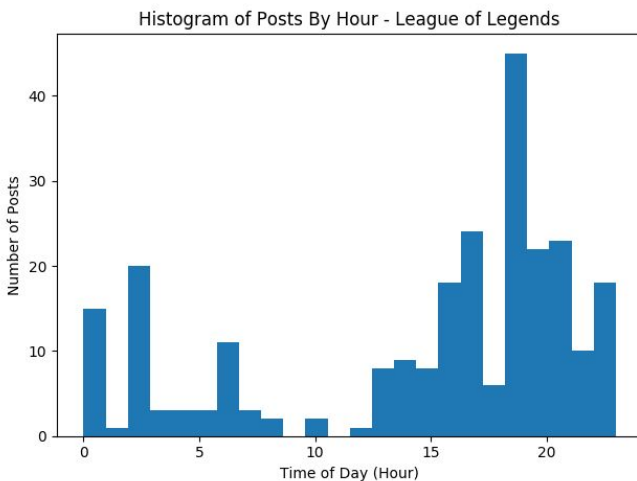
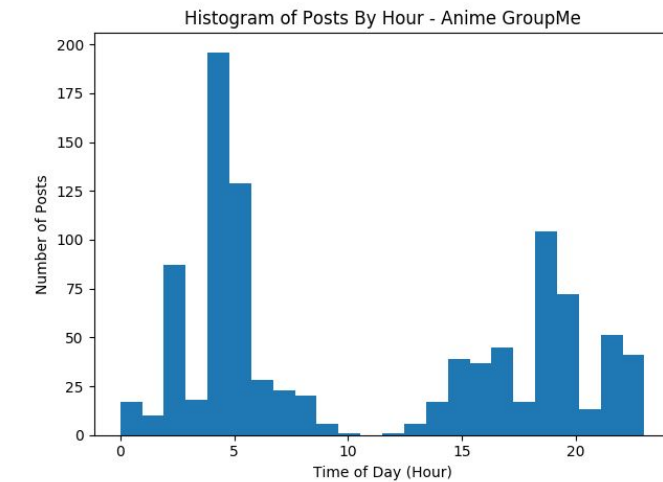
To interact with the API, an application called *Discord Chat Exporter* was utilized to pull chat logs, along with the date and times at which they were posted. These logs were saved in plain text files and parsed using Python.

C. Analysis Method

There were several metrics taken into account for the analysis. The data was first scoped purely to the month of November. This was done by querying database entries between midnight on November 1st and 23:59 on November 30th. This trimmed the data and established a baseline dataset to work with.

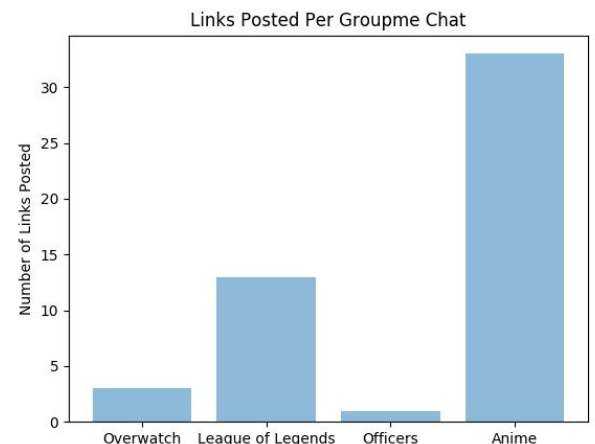
The first metric analyzed is time of posting. In the interest of seeing when people are most active, a

query was done where the data was sorted by hour of the day. This was then interpreted as a histogram and broken up into 24 columns.



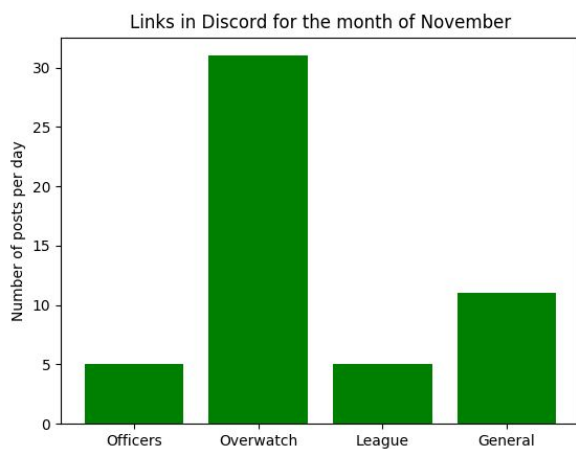
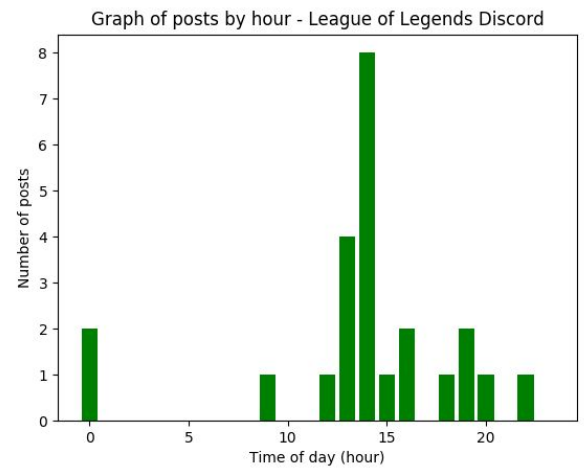
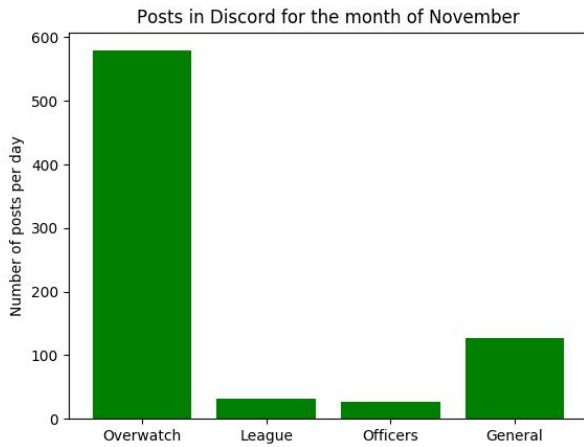
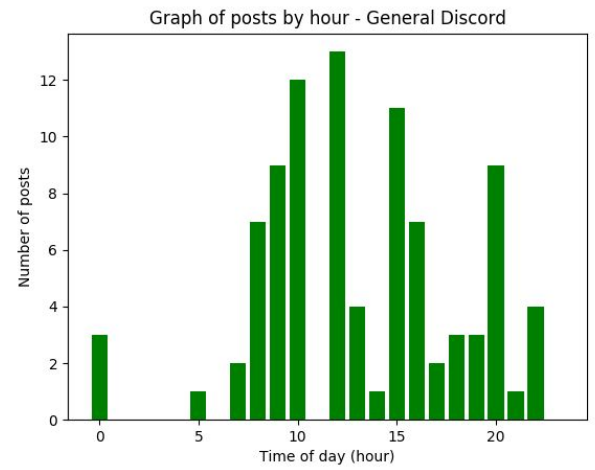
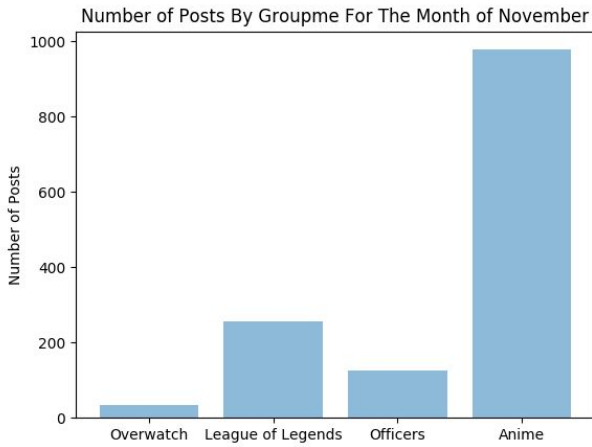
Further analysis involved a basic word count. All the text fields from all posts from a chat or discord channel were split into words and then added to a word count dictionary. Stop words were manually generated by manually examining the top 100 most common words from a chat and stripping out words that were deemed to have little descriptive value such as pronouns and articles.

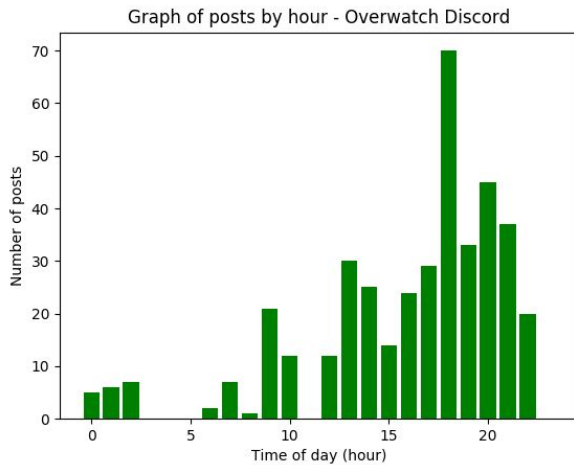
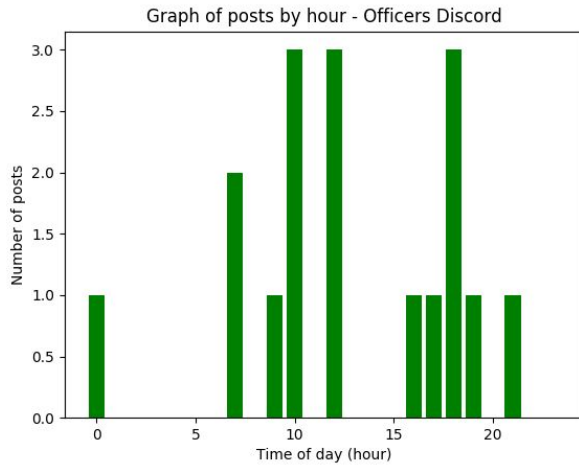
Taking that one step further was analyzing the text fields for links. This was performed rather simply by searching for the “https” substring in each text field. The counts were then divided by the total number of posts for November to give a percentage. This data is given in the presentation.



Additionally, a basic count of the posts in each group was done. This simply counted the number of entries in every chat query. This basic count provided a baseline value to compare the links

posted and word frequencies as a percentage of the whole.





IV. RESULTS

Most chats experienced a spike in usage around 6:00pm, however all GroupMe chats, especially the Anime group, showed a second spike around 3:00am. This is due to late night users posting in these groups. Discord does not show this second spike as much.

Additionally, the Anime GroupMe and the Overwatch channel in Discord showed the most activity, both in links and in gross number of posts. These two chats have achieved a critical mass of users and see regular activity while the other chats see a fraction of the activity.

It is noteworthy that the officer chats in both GroupMe and Discord are not very active, although

the Discord Officer channel appears to use links slightly more than its GroupMe counterpart. This may reflect an unconscious shift in communication medium among the officers.

In the word frequency charts (in the presentation), it is shown that while all the chats seem to discuss the topics relating to their titles, League of Legends discussion makes it in everywhere as seen by references to the ranking system via “gold” and “silver” or “champions”.

V. CONCLUSIONS

GroupMe and Discord users have different usage patterns, even for the same subject matter. Overwatch is more popular on Discord, and League of Legends is more popular on GroupMe. Overwatch chatters on both applications have similar time of day posting patterns. Officers on GroupMe post almost no links, while officers on Discord was examined in the same way as GroupMe. Discord have 18.5% of their posts containing a link. Overwatch Discord users post more about professional gaming than their GroupMe counterparts. Users in general post more links on Discord.

Some of this may be due to the origins of the club. Originally the League of Legends Club, all the club communication was hosted on GroupMe and the change to Discord was accompanied by the simultaneous expansion to other esports as well as the advent of Overwatch as a game.

VI. FUTURE WORK

Future work would include implementing categorizations of phrases. Seeing the frequency of words in relation to the words they are commonly posted with would provide deeper insights into how users communicate. Gathering picture data would also add another layer of information about usage patterns. If we could analyze when pictures are posted, we could see when users typically prefer to communicate via visual means.

REFERENCES

- [1] M. Arrington, "GroupMe, Born At TechCrunch Disrupt, Secures Funding And Launches," *TechCrunch*, 25-Aug-2010. [Online]. Available: <https://techcrunch.com/2010/08/25/groupme-born-at-techcrunch-disrupt-secures-funding-and-launches/>. [Accessed: 04-Oct-2017].
- [2] Takahashi, Dean (September 10, 2015). "Hammer & Chisel pivots to voice comm app for multiplayer mobile games". VentureBeat.
- [3] Marks, Tom (May 14, 2016). "One year after its launch, Discord is the best VoIP service available". PC Gamer. Future plc.
- [4] Dev.groupme.com. (2017). *GroupMe Developers*. [online] Available at: <https://dev.groupme.com/docs/responses> [Accessed 4 Oct. 2017].
- [5] Dev.groupme.com. (2017). *GroupMe Developers*. [online] Available at: <https://dev.groupme.com/docs/v3> [Accessed 4 Oct. 2017].
- [6] "Discord API Docs for Bots and Developers," *Discord*. [Online]. Available: <https://discordapp.com/developers/docs/reference>. [Accessed: 04-Oct-2017].
- [7] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [8] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/>
- [9] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
- [10] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [11] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [12] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [13] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.