Discourse on Discord: An analysis of relations between users

SILAS HOEVERS, University of Twente, The Netherlands

ABSTRACT

Discord is a platform for online communication. While its original audience mostly consisted of people playing video games, a shift took place in its usage during the COVID-19 pandemic. Nowadays the same application is used by book clubs, students and teachers and other communities. But what role does the platform play in connecting these individuals? The versatility of the platform means people can communicate in a multitude of different ways. So how do people use the platform to reach each other? This paper tries to give an insight into the relations between users within different communities. A tool is created to extract data from the platform after which it is used to create an overview of the nature of connections between users. This data is then used to look at the usage of different mechanisms to enhance messaging. Where mechanisms like mentions and replies to messages are used quite extensively, other special mentions like @here and @everyone are sparingly used. But why is this? In this paper we try to give explanations for this phenomenon, providing an insight into how users connect on the platform.

Additional Key Words and Phrases: Discord, monitoring, OSN, relations, bot

1 INTRODUCTION

The past few years has forced a lot of our communication to take place online, due to limitations in the amount of physical contact that was allowed thanks to COVID-19 regulations [8]. This move has changed the way we communicate. While trying to overcome the shortcomings of communicating online ([13]) we have developed other means to convey aspects lost, like emotion and intonation. In the digital space we still make use of text as a means to communicate nowadays, even though this means of communication provides the bare minimum for communicating between individuals. However, online social networks (OSNs) have increased the amount of tools a user has at their disposal to add more expressiveness to their messages. This is quite the step up compared to the original way we used to send texts by SMS, having purely ASCII characters available to us to add a simple smiley, enabling us to convey very simple emotions like sadness or happiness.

Examples of these types of tools are the ability to embed images or video or responding with so called *emotes* to messages. Next to that, mentioning or replying to other users is possible.

These tools are a requirement in an environment where a single channel meant for text messages is used by multiple users at the same time. Even though designated channels can be used to separate conversations by their subject matter, without the use of mentions and replies these conversations would become hard to follow as multiple people are interacting with each other within the same channel.

With this broad range of tools at their disposal and the large amount of freedom users enjoy online, upholding social norms becomes more difficult. Where a small community might purely rely on peer pressure to ensure users follow a set of (unwritten) social rules, bigger communities install so called *moderators* in their servers. These individuals are tasked with upholding social norms, removing content or even users themselves that are in violation of said norms. A common theme is that communities explicitly post the guidelines that users have to follow within the community. Of course different communities formulate different types of guidelines, some being more strict than others.

With a total active user count of around 150 million people [5], Discord is one of the largest OSNs out there. Discord hosts so called *servers* or *guilds* for a wide range of different communities. Where originally these communities were mostly centered around a culture of video games, the COVID-19 pandemic caused an influx of users from other types of communities [11].

These servers are how Discord distinguishes itself from other online social media networks like Facebook or Twitter. Instead of having a single wall that every user can post their own content to, a user has to intentionally join a specific community. Servers in Discord act as a kind of *Forum Romanum*. Users can communicate through different channels within a server. How these channels are set up differs per community. Servers used for playing video games will have voice channels that can be used to sit together and discuss strategy while an educational server will have a wide range of text channels that can be used by students to post questions to the teaching staff or other students.

Where most servers will not exceed a user count of 500, others pass the 250K mark [12]. While moderating a community of a small size is still manageable by hand, this becomes an impossible task with bigger communities. To keep up with the scale of such a big community, automated tools for moderation can be used. These tools can be given different levels of responsibility. At the most basic level, a tool like this can flag messages or users if they are considered to be in violation of community guidelines. Manual verification still needs to take place in that case. Giving more responsibility to a tool like this will entail the automatic removal of content and/or users.

Discord provides an API, enabling developers to write so called *bots*. These bots can be interacted with by users or can passively execute the tasks they were designed to perform. Examples of usages of these bots are listening to music, providing motivational quotes or gamifying interactions on a server.

In this paper we try to lay the groundwork for the monitoring of the behaviour of users, with the intention that it could be used as part of the design for some kind of bot that deals with users expressing negative behaviour. We will focus on the relations between users, analyzing which mechanisms users tend to use to connect with each other.

TScIT 37, July 8, 2022, Enschede, The Netherlands

2 BACKGROUND

Some technical background has to be provided before we formulate our research questions. In this section we try to provide enough information to help the reader understand the questions posed while trying to not burden them with too much details.

Discord as an application is quite new, with its first release in 2015. Once a user has created an account with Discord they are able to join servers and connect with other users. Access to a server is handled by means of invitation. Users can be directly invited or be provided an invite link to a server that they can use to join.

Discord facilitates VoIP (Voice over IP), enabling users to talk to each other by voice over the internet. Next to that it supports instant messaging. This division is made possible by the ability to create specific channels where each channel is either meant for voice or for text:

- *Voice channels*. Once a user has joined a voice channel they are able to communicate by means of their microphone and they are able to hear what other users in this same channel are saying.
- *Text channels*. Users can post messages (possibly paired with images or videos) within these channels. Users can reply to specific messages of other users as well as include usernames of users. The latter is called *mentioning*.

Within this research we will focus on text channels as they are less ephemeral and easier to interact with through the API.

A unique feature that Discord supports is the assigning of roles to users. Next to providing a potential social status these roles also serve a functional purpose. Distinctions can be made between users on the same server by means of these roles, enabling access to channels that are locked off to regular users. An example would be the role of *@Teacher*, enabling access to chats discussing the performance of students.

Users are uniquely identifiable by means of their personally chosen username and a 4-digit prefix acting as a discriminator. This way users are free to pick whatever username they want without being constrained by uniqueness. In the background, however, users are given a separate unique ID, which we will be using to identify users.

2.1 Mentions & Replies

In this research one of the focuses will be the act of mentioning other users or replying to their messages. Mentioning other users means that we include their username in our message, turning it into a clickable reference to their account. When we reply to a message we include a reference to the original message in our new message. This way users can easily see what message the reply is about.

Both of these mechanisms serve the purpose of drawing extra attention from other users. Perhaps the other user that we are referencing has the channel that we are posting in or even the whole server *muted* for themselves, meaning they will not receive any notification about the new message that we are publishing, unless the mention or reply is included. By means of a mention or reply we can ensure the other user is properly notified about the fact that we are talking about them or their message.

2.2 @everyone and @here

Apart from mentioning other users there also exist special types of mentions that do not point to any specific user, but have the same aforementioned effect. The special types of mentions that this research focuses on are *@everyone* and *@here*. While @everyone will notify every single user in the server, @here will only notify those that are online at the time of posting the message.

3 GOALS

To help understand the behaviour of users we will look at the relations between them. There are multiple hurdles to overcome before we can determine these relations, however. The first problem is gathering data about these relations. If we want a data set that is large enough to support our claims we will not be able to manually register all possible messages we want to use for our analysis. Especially when we add ourselves to bigger servers that generate heaps of traffic this becomes an impossible task. This means we have to automate this process.

Another problem is how we determine whether an actual relation exists between two users. In the previous sections we have discussed a selection of tools available to users, specifically those provided by the Discord application. But are users actually using these tools to enhance their messaging experience? Perhaps users prefer different methods to connect with each other?

With this paper we try to answer the following research question.

What relations exist between users on Discord?

To help us in answering this question and make the goal to be achieved more concrete we have formulated three (sub-)research-questions.

RQ1: How can a bot be built to extract user generated content on Discord?
RQ2: To what extent are special mentions like @everyone and @here used?
RQ3: To what extent do users interact with each other by means of replies and mentions?

4 DATA ACQUISITION (ANSWERING RQ1)

In this section we try to answer the first research question posed. We do this by describing what different stages were involved in the development of the bot, including initial requirements and extra considerations in terms of anonymization of data and the like.

4.1 Requirements

Before the process of development of any piece of software we have to settle on what requirements the application has to adhere to. These requirements can describe clear features but also certain quality assurances like timing or performance. For the development of the bot that will be used to gather data we have determined the following set of requirements that it must comply with:

- The bot has to be able to run 24 hours a day, 7 days a week, without interruption.
- The bot has to be able to execute its tasks without the necessity to be added by the admin of the server.

- The bot has to be able to receive an event every time a message is sent.
- The bot has to have access to the content of these messages to be able to determine the presence of special mentions.

It has to be noted that during the process of data collection, we did not make any distinctions between whether a message was a reply or mention. We only noted the other user that was involved in the exchange.

4.2 Design of the bot

By means of the *discord.py-self* library we can define the behaviour of our bot in abstract terms while the handling of the connection is performed in the background.

While setting up the connection with the Discord server we have to pass a token identifying who we are. We were able to extract our token by intercepting and dissecting packets from a connection made by an original Discord client. This token is saved in a separate *.env* file for security reasons. By separately importing this token as an environment variable we do not have to store the token in the code, potentially exposing it to the outside world.

Once the connection has been established we can receive and act upon events within the *on_message* function. When we receive such an event we take the following steps:

- Check whether the event is actually a message (and not a different type of event like a user joining a server) and whether this message was sent in a server (and not a direct message). Since we are only concerned with messages upon their creation, we filter on the MESSAGE_CREATED event.
- Extract the location of the message (server and channel IDs).
- Extract the author of the message and anonymize their ID.
- Check if any other user was mentioned or replied to, extract these users and anonymize their IDs.
- Check if any other attributes were included in the message like special mentions or media and add their respective flag if this is the case.
- Attach a timestamp.

The collected data is stored in a CSV file for easy retrieval at a later time.

4.3 Anonymization and Privacy

Because we are dealing with user generated content that is privacy sensitive we have to anonymize our results. Be means of hashing we apply a difficult-to-reverse mapping to the information we collect such as the user ID. Because computing a hash always nets the same result we are able to both anonymize the data and retain existing relations between different users.

The hashing algorithm used is part of the SHA-2 set of cryptographic hash functions. We make use of the SHA-256 variant. The implementation of this function finds its origin in the *haslib* library. For every identifier that the hash is performed on the following steps are taken:

(1) Cast the integer to a string-type
(2) Encode the string to pure bytes
(3) Create a hash object with the desired algorithm with the encoded string as input
(4) Perform the actual hashing by requesting the hexadecimal representation of the *digest*

All these steps together are performed by the following line of code:

```
hashlib.sha256(str(id).encode()).hexdigest()
```

The hashed IDs are stored alongside the rest of the message data.

4.4 Expansion of data collection

To ensure we collect a sufficient amount of data we joined 7 servers that are considered to be the most popular, with some being among the largest in the world. In the rest of the paper we will make reference to them by a number to increase the anonymity of the data. To support our analysis we do however provide some general descriptions of these servers:

The server is a community surrounding...

(1) ... an online personality
(2) ... a video game
(3) ... the desire to study together
(4) ... the desire to discuss the financial market
(5) ... a video game
(6) ... the desire to connect with other people with romantic intent
(7) ... an online personality

4.5 Data collection

The actual process of data collection was performed on a separate Linux server. This server was running Ubuntu Server 20.04. By means of the *nohup* (No Hang Up) command we were able to start the script and close the terminal connected to the server without terminating the script. To sync the data between the server and the machine that the analysis would be performed on, a private Github repository was used to commit all data that was collected to. This was manually done on a daily basis. The repository doubled as a kind of remote backup this way.

5 RESULTS

In this section we given an overview of the data set we have collected. Next to that we try to answer our remaining two research questions using this data.

5.1 Data set

A total of 859817 messages were collected between the 6th and the 11th of July 2022. These messages originate from the 7 bigger servers we added ourselves to plus 7 other servers that we were already a member of. Due to the fact that these other smaller servers produced little traffic, yielding them irrelevant for our research, we tend to exclude them from our analysis from now on, except when comparing large and small servers is deemed relevant. An overview of the distribution of messages over the different servers can be found in figure 1. When we plot the amount of messages sent, clustering them on intervals 1 hour in length, we see a graph appear with a wide array of peaks and pits. The eb and flow is illustrated by figure 2.

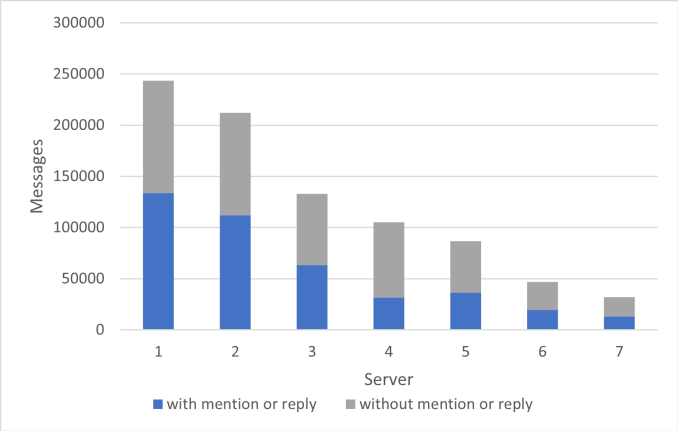


Fig. 1. Distribution of messages with and without either a mention or reply among different servers.

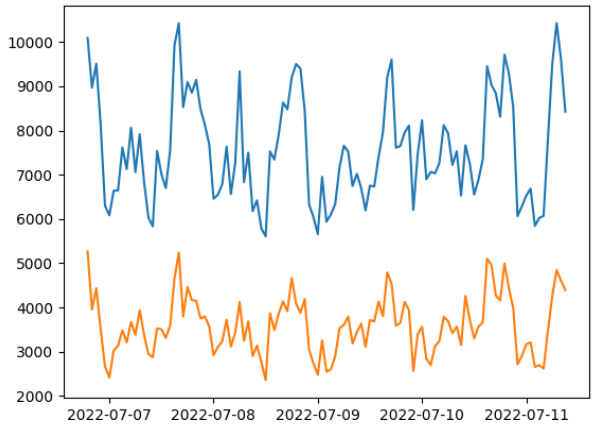


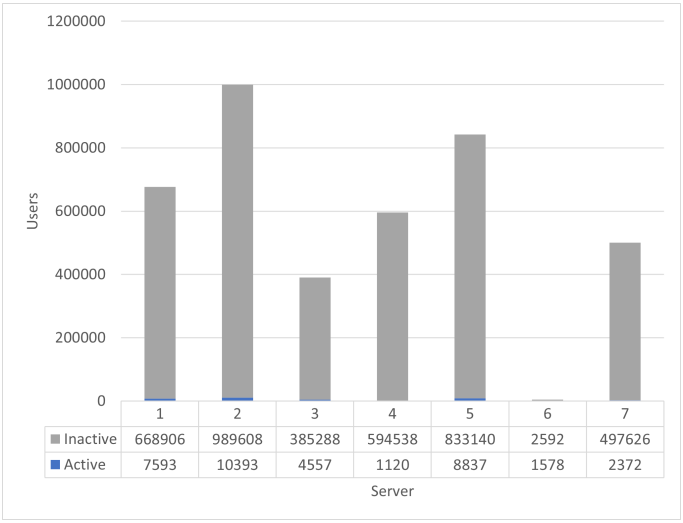
Fig. 2. Distribution of messages over time, clustered per 1 hour. Blue is the total amount of messages, orange represents the messages that included a mention or reply

The 7 most relevant servers to our research vary significantly in size. It has to be noted that the maximum number of users within a single server is 1 million users, which is reached by server number 2. This means users trying to join this server right now are greeted with the message that the server is at capacity. An overview of the distribution of users between servers can be found in figure 3.

It has to be noted, looking at figure 3, that only a very small portion of users registered to a server partake in conversations. We consider a user to be *active* if they were seen posting at least 1 message in a server.

5.2 Answering RQ2

At a first glance the results seem very clear cut. Of the more than 800.000 messages collected, only 35 of them included either an



	1	2	3	4	5	6	7
Inactive	668906	989608	385288	594538	833140	2592	497626
Active	7593	10393	4557	1120	8837	1578	2372

Fig. 3. Distribution of active (has been seen sending at least 1 message) and passive (not seen sending a message) users among different servers

@everyone or @here mention. Table 1 illustrates the distribution of these messages among our bigger servers (1 message was excluded as it was within one of the smaller servers). Most of these messages were sent in channels where regular users are not permitted to post in. Think of channels that are meant for announcements for example. These announcements were used to for example let users know that there were changes in the community guidelines or, in the case of servers surrounding an online personality, that there were notable events happening surrounding this individual like the posting of a new video or stream on another platform.

The claim above seems to be supported by the fact that of the 4 messages including an @everyone mention, 3 of them were accompanied by an embedding, meaning external content like a URL was included within the message. All of these 3 messages were posted in channels meant for announcements regarding the posting of external content. 2 of these messages were sent by the same user, but in different servers. It is quite likely that these messages were sent by a bot, used to automate announcements. Such bots can be shared amongst different servers, while they retain the same "user" ID. No distinction is made between regular and bot users in that regard.

It seems the usage of special mentions like @everyone and @here are very limited. We base this on the fact that only very little messages in our final data set include such types of mentions.

Due to fact that these special types of mentions have the ability to reach so many users within a single message, circumventing any muting that might be enabled by a user, they tend to be abused as a form of spam. Thanks to this, *administrators* (individuals that have control over the settings of a specific server) have the ability to disable the usage of such mentions or constrain their usage to certain individuals (identified by a role). The individuals that are allowed to post messages with these types of mentions presumably only make use of them when truly necessary as in the case of the announcements mentioned before. Certain individuals can be given

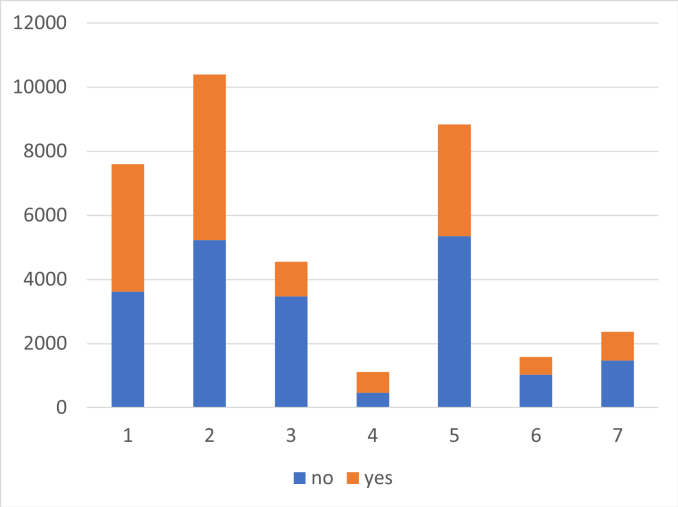


Fig. 4. Active users posting at least one message with a mention or reply. "No" if an active user was never seen using a mention or reply, "yes" if they we seen using at least one mention or reply

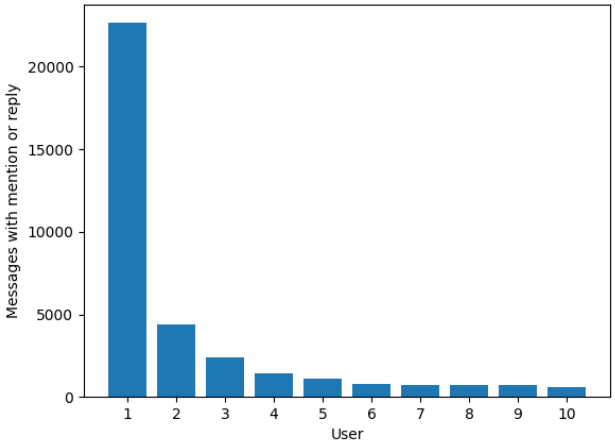


Fig. 5. The distribution within server number 3 of the amount of messages posted with a mention or reply by the 10 users who post the most messages with this mechanism

a role enabling them to post in these channels locked off to regular users.

5.3 Answering RQ3

When looking at figure 1 we can see what portion of messages included either a mention or reply, while table 1 gives the exact numbers. Most bigger servers sit close to 50% of its messages including either a mention or reply, with the exception of server number 4, with 30%. Another metric in which this server differentiates itself in is the amount of messages per active user, with almost a 100 messages, compared to other servers that do not cross the 40 message threshold (as can be seen in table 1).

The outlier that server 4 is among the bigger servers can be accounted for by the fact that multiple channels in this server are dedicated to not necessary conversation as much as one-directional posting. An example would be the fact that server 4 contains a channel that is used for counting up from 1. Users simply post the next number in the sequence and if the sequence breaks, the counting starts over. Looking at this channel alone the ratio of messages with a mention or reply against the total number of messages is just below 0.002 (0.00002%).

To illustrate the difference between larger and smaller servers, we can take two of the smaller servers previously excluded from our analysis. The distribution of their messages can be found on the bottom of table 1. Both servers are used in an academic context, with server 8 being used primarily by students to connect informally, while server 9 is used by both teachers and students in a more formal manner regarding a specific set of courses. Both servers show a smaller percentage of messages with a mention or reply compared to bigger servers.

Among *active* users we see variable amounts of use of the mention or reply mechanism. As mentioned before, active users are users that have been recorded posting at least one message. Figure 4 shows the distribution of active users that have used the mechanism at least once against those that have never used it.

When we try to get a sense of how much the mechanism of mentioning and replying is used among users within the same community, we see an interesting picture appear once we zoom in on our different servers. If we focus on the 10 users that post the most messages that include a mention or reply, most servers have a steady drop off. However, if we look at the distribution of server number 3, visible in figure 5, we see the first user posting significantly more than the rest. Our assumption is that when there is a user posting significantly more than the rest, this hints to some kind of automated behaviour, meaning that this user is probably a bot. Upon manual inspection this seems true as we can pick out a particular channel in this server that is used purely to mention other users. This is done by a bot, set up to comment about the behaviour of users that might be considered rude like keeping their camera turned off while in a voice channel. This has the effect that every single message sent in this channel is accompanied by a mention.

While looking at the time distribution in figure 2 it appears there is no significant difference in how messages are distributed across time depending on whether they include a mention or reply. The same peaks can be identified in both plots.

Depending on the size of the community the necessity to use mentions and replies seems to grow with the size of the server. Of course, depending on how the channels are set up, conversations can be dispersed more evenly among different channels. In a server with a high user count and only a handful of chats available, the use of mentions and replies seems a must, whereas smaller servers with more channels that can be used for discussion can keep conversations clearly apart without the need for these tools.

Server	Total messages	Messages per active user	Messages with @everyone or @here	% of total	Messages with mention or reply	% of total
1	243584	32	6	0.002	133658	54.87
2	212224	20	0	0	111865	52.71
3	132785	29	0	0	63261	47.64
4	105264	94	2	0.002	31509	29.93
5	86736	10	17	0.020	36529	42.12
6	46697	30	1	0.002	19427	41.60
7	31922	13	8	0.025	13096	41.02
8	489	15	0	0	52	11.90
9	164	5	0	0	33	23.74

Table 1. Number of messages per server including the distribution of special mentions and messages with a mention or reply. Servers numbered 8 and 9 are included for reference.

6 ETHICAL CONSIDERATIONS

As mentioned before, this research deals with personal data of users, be it limited. Steps are taken to anonymize the data so that if the data set is leaked the results are not reversible. However, we are technically in violation of the Discord Developer Policy [4]. This policy defines *Discord Data* as "any and all data you obtain through the APIs". It is stated in the policy that it is disallowed to collect data that is not used for the functioning of your application for Discord. This puts our research in a legal grey area. As a resolution it was agreed upon that the data used for our analysis would be deleted from every location it was stored, while the code would remain to retain the reproducibility of the study.

Another problematic aspect of this research is that the answering of RQ1 resulted in the development of a so called *self-bot*. A self-bot is considered to be the automation of actions on behalf of a user account (and not an officially registered bot). This creates the risk of termination of the account used within the research.

It has to be mentioned that some of the privacy concerns raised above are being addressed by Discord. As of August 2022 it is not possible to access message content by default and permission has to be granted before a bot has access to attributes related to content [14]. Interestingly Discord makes a distinction between bots that are more widely adopted, only enforcing this limitation if the bot in question is part of at least 75 servers.

7 RELATED WORK

A broad range of existing research was taken into account in the writing of this paper. The common theme between these different papers are OSNs. Some of the research is generalised, while other papers describe an analysis of specific platforms. Most technical analysis regarding the Discord application was forensics, trying to give a sense of what data can be captured off of a local installation of the application. Our technique of data collection is done through more official channels, using the official Discord API. This also means our focus lies more on the application level instead of the machine level.

7.1 Performing forensics on the Discord application

Existing papers on the Discord application and its security aspects specifically perform a technical analysis on the potential artefacts the application leaves behind on a system. Iqbal et al. developed a tool for extraction of cached data [6]. Moffit et al. built on top of this analysis and focus specifically on the Windows 10 version of the application [10]. The common denominator of this existing research is that all analysis is performed after the fact. Instead of looking at how communication between users envelops over time, a single point is taken after which all logs are dissected. To be able to extract information at a rate that enables us to say something meaningful about relations between users developing over time we should be able to receive messages as they are perceived by any other user. This means producing a close to live feed of what is happening on a specific server.

7.2 Applications of Discord and their effects

As mentioned before, due to its newly found audience in recent years Discord is not only used for communication during video games anymore. We can see the same platform being used in for example the field of education. Bridson et al. gives a report on their experience with using the application as part of a course on software engineering [2]. A very extensive report on the abuse of Discord and a comparison with its predecessor, TeamSpeak, have been developed by Bryant [3]. A qualitative analysis on the experiences of moderaters has been performed by Jiang et al. [7]. This research showcases different strategies for dealing with user (mis)behaviour and how dealing with this behaviour differs compared to other OSN.

7.3 Monitoring / Analyzing user behaviour on OSN

A large body of research related to the behaviour of users exists, but it is generalised towards Online Social Networks. Examples of these networks are Facebook, Twitter and Reddit. Due to the nature of organisation that can take place within a single Discord server we thought these works relevant as well. Kou and Gui performed an analysis on the automated flagging of toxic behaviour in the video game called League of Legends [9]. Where the ability to flag the posts of users or even users themselves are visibly present in the previously mentioned OSNs, these features are not so prevalent in Discord. How exactly the automatic flagging could be performed is described by Alsharef et al. [1]. Thanks to this report we are also reminded of the fact that different levels of misbehaviour can be identified. Where we initially might see things in terms of black and white there seems to be a more complex story at hand.

8 DISCUSSION AND LIMITATIONS

In this section we would like to mention a few points that can be used as a kind of behind the scenes, giving an insight into what other things were tried and failed. Where this was considered by the researcher some points are also added about how things could have been different in the process of the research. Considering the quality of the research and the limited time-frame that was given for the execution of the research certain proverbial corners were cut. In the potential future a rerun or expansion might be considered and we would like to give the individuals taking this task upon themselves a head start.

8.1 Hidden channels

It has to be noted that not all messages that are posted in a server are collected. As mentioned before, by the means of special roles a distinction can be made between different users. There are most probably channels within a server that we do not have access to that could change the shape of our results if they were included.

8.2 Security

A note has to be made about the security risks with the methodology used. The token that is used to connect to Discord uniquely identifies the user (or bot). If this token is intercepted by a party with malicious intent it can be used to steal the identity and use the account of the user associated with the token. Changing the password of the account associated with the token mitigates this as it forces the token to be regenerated. In this research extra care has been taken to prevent the token of the account used falling in the wrong hands.

8.3 Future

As was mentioned before in Section 4, we did not make any distinctions between whether a message was a reply or mention during our data collection. Perhaps during a future run of the same experiment these two types of information can be separated, painting a different picture than the one this paper gives.

Next to that we assume that the inclusion of the actual content of messages sent would be a valuable addition to the data, keeping in mind the ethical risks this addition brings along with itself. If we want to work towards a system that can automatically detect misbehaviour or even targeted negative attention, we can apply a sentiment analysis on this content to determine the nature of the messages themselves.

Of course the expansion of data collection can always make a research more rigid. More servers of varying types of communities could be added. Within this expansion one should also include smaller servers to reduce bias towards bigger communities.

One hurdle that has to be overcome with this approach though is the pace at which data is collected, minding the connectivity, speed of processing and storage that might bottleneck the data collection process once more traffic is being collected.

9 CONCLUSION

To wrap up we would like to look back at our original research question and the results to the sub questions. Our main focus was identifying relations between users, based on certain special properties of messages that we collected.

In the development of our bot we went through multiple iterations and versions, ending up with a version that was deemed significantly more stable and reliable than previous versions. Switching from a manual implementation to one using a library caused the most significant change in that regard. Where earlier versions of the bot tended to not behave very nicely regarding keeping the connection alive, introducing a library to help with this process was what we needed in the end to get a proper flow of data going. With little coding knowledge we were able to construct a functioning listener, intercepting every message sent within the servers that we are a member of. Our concern regarding how easy it was to develop such a bot has been addressed in the *Ethical Considerations* section 6.

Using the data that was collected by the bot we went about answering our other two research questions. Firstly, the data paints the picture that the usage of the @everyone and @here special mentions was very limited. We have reasoned this is most likely due to the fact that these types of mentions are sensitive to abuse from users, resulting in their usage being blocked most of the time. Thanks to this, these types of mentions are not used by regular users and tend to only appear when users with higher privileges want to reach everyone on a server with important announcements. Secondly, we seem to find variable usage of the mention and reply mechanisms among different servers/communities. In terms of the extent of their usage we seem to find that just below half of all messages sent in a server are accompanied by a mention or reply. It seems the results point to a distinction between larger and smaller servers in that sense, where there is a higher necessity to use mentions and replies in bigger servers as conversations are otherwise harder to conduct in channels used by multiple users at the same time.

10 ACKNOWLEDGEMENTS

I want to thank Etienne Khan for his support as my supervisor during the performing of this research. He always had answers to my questions and stuck it out with me until the end, never losing faith in my ability to make this research a success. Next to that I have to thank Anna Sperotto for the exceptional help she has given me during the more stressful periods of this research.

REFERENCES

[1] Ahmad Alsharef et al. "An Automated Toxicity Classification on Social Media Using LSTM and Word Embedding". In: *Computational Intelligence and Neuroscience* 2022 (2022). ISSN: 16875273. DOI: 10.1155/2022/8467349.

[2] Kathryn Bridson, Jeffrey Atkinson, and Scott D Fleming. "Delivering Round-the-Clock Help to Software Engineering Students Using Discord: An Experience Report". In: (). DOI: 10.1145/3478431. URL: https://doi.org/10.1145/3478431..

[3] Oliver Bryant. "Retrieval of Digital Artefacts from TeamSpeak and Discord: A Forensic Investigation and Analysis of the Malicious Use of Gaming Communication Clients. Evaulation of IoT for Wellbeing View project Retrieval of Digital Artefacts from TeamSpeak and Discord: A Forensic Investigation and Analysis of the Malicious Use of Gaming Communication Clients. View project". In: (). DOI: 10.6084/m9.figshare.12656990. URL: https://www.researchgate.net/publication/320290534.

[4] *Discord Developer Policy*. URL: https://discord.com/developers/docs/policies-and-agreements/developer-policy.

[5] *Discord: About Us*. URL: https://discord.com/company.

[6] Farkhund Iqbal, Michal Motylinski, and Aine MacDermott. "Discord Server Forensics: Analysis and Extraction of Digital Evidence". In: *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, Apr. 2021, pp. 1–8. ISBN: 978-1-6654-4399-9. DOI: 10.1109/NTMS49979.2021.9432654.

[7] Jialun Jiang et al. "Moderation challenges in voice-based online communities on discord". In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (Nov. 2019). ISSN: 25730142. DOI: 10.1145/3359157.

[8] Simon Kemp. *Report: Most important data on digital audiences during coronavirus*. URL: https://thenextweb.com/news/report-most-important-data-on-digital-audiences-during-coronavirus.

[9] Yubo Kou and Xinning Gui. "Flag and flaggability in automated moderation the case of reporting toxic behavior in an online game community". In: *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery, May 2021. ISBN: 9781450380966. DOI: 10.1145/3411764.3445279.

[10] Kyle Moffitt et al. "Discord Forensics: The Logs Keep Growing". In: *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, Jan. 2021, pp. 0993–0999. ISBN: 978-1-6654-1490-6. DOI: 10.1109/CCWC51732.2021.9376133. URL: https://ieeexplore.ieee.org/document/9376133/.

[11] Monica Chin. *Discord raises $100 million and plans to move beyond gaming*. URL: https://www.theverge.com/2020/6/30/21308194/discord-gaming-users-safety-center-video-voice-chat.

[12] Potato. *Server Member Cap Increases*. URL: https://support.discord.com/hc/en-us/articles/360052841734-Server-Member-Cap-Increases-.

[13] Lauren E. Sherman, Minas Michikyan, and Patricia M. Greenfield. "The effects of text, audio, video, and in-person communication on bonding between friends". In: *Cyberpsychology: Journal of Psychosocial Research on Cyberspace* 7.2 (July 2013). ISSN: 1802-7962. DOI: 10.5817/CP2013-2-3.

[14] Weh. *Message Content Privileged Intent FAQ*. June 2022. URL: https://support-dev.discord.com/hc/en-us/articles/4404772028055.