Department of Computer Science

Submitted in part fulfilment for the degree of BSc.

How does chat engagement vary between the audiences of different streamers, and why?

Matt Geneux

May 2022

Supervisor: James Walker

Contents

Executive Summary

The aim of the project was to investigate how and why chat engagement varies between the audiences of different streamers on the website Twitch.tv. The approach chosen was to separate streamers based on the genre of game they were playing, a relatively unexplored approach, with related works tending to differentiate streamers by their audience size instead [1], [2]. This approach therefore provides an opportunity for new insights to be uncovered, which are unknown before the completion of this project. The chosen genres were the somewhat general terms "shooter" and "non-shooter" which were chosen for their ability to split the dataset into a relatively even 60/40 split.

Understanding how different audiences engage with livestream chat could potentially unlock numerous benefits for a number of different groups, including Twitch, advertisers and users. This makes it an important problem to study. For Twitch, like any large company, understanding how their users interact and engage with their platform is priceless. Twitch benefits financially from having a large active user base at any given time, making their money primarily through subscriptions and in-app purchases [3]. It stands to reason that having more active users on the site will help sell more subscriptions and in app purchases. Therefore, noticing trends in how users engage with the site can help Twitch steer content in a direction that will keep these users active and in turn help them see a financial benefit. For advertisers, understanding user habits is beneficial across any platform. If they can work out what interests a user, they can target advertisements that are more likely to draw the user's attention [4]. Therefore the user is more likely click on the advertisement and potentially purchase what it is being sold. For users, their overall experience of the platform can be improved, even if this is just because of the previous two factors. Adverts related to their interests theoretically should be more appealing. Whilst targeted content which relates to their interests will streamline their experience with the site, potentially eliminating the need to scroll through lots of channels to find the content they want to watch.

The methodology for the solution centred around the implementation of a machine learning model, using the data set provided by the TwitchChat paper [5]. Specifically, by using the Latent Dirichlet Allocation (LDA) algorithm, four topic models would be fitted. The input data would be live stream chats

relating to two ‘shooter’ games, and two ‘non-shooter’ games. LDA works by dividing a given document up into n chosen topics and grouping related words together [6], [7]. This then allows for a visualisation of the topics in the model, which could appear more or less frequently throughout the duration of a given stream.

Chat engagement was measured by looking at both topic distribution throughout each chosen stream, and the nature of the topics in each stream. A combination of both methods allowed for a visualisation of the peaks in audience activity, and an analysis of the nature of these peaks. This lead to a better understanding of what drives the interest of the audiences of different streamers, by showing what topics of conversation spark increased activity in the livestream chat. The topic distribution is a measure of how evenly the topics are being represented at any given moment in the stream. This was measured by calculating a value known as the Gini coefficient, which is primarily used in economics to measure income inequality [8]. Using methods inspired by [9], this function returned a value between 0 and 1 to show how evenly (or not) the topics were distributed. To support this, the topic percentages were plotted for the selected streams, which displayed a visualisation of how each topic increased and decreased in prevalence throughout the duration of each stream.

This method provides a good solution to the identified problem. It allows for two distinct methods of analysis which can be combined to give a clear understanding of how different audiences engage with their respective livestreams. By extracting the topics from the livestreams, we can see how these are distributed throughout each stream. Furthermore we can interpret the nature of these topics to get a better understanding of the mood in the chat at certain periods, and perhaps make judgements of what was happening in the stream at a particular time.

The results show that the non-shooter games have less variety in chat content, with a less equal topic distribution. Furthermore, there is a clear correlation between the nature of the topics that cause peaks in chat engagement and the genre of the game being played. There is seemingly a slightly more hostile tone to the shooter chats, with some peaks of potentially hostile activity with chat participants labelling other players as "sweaty" and "try hards". This activity is seemingly less apparent within the topics of the non-shooter games. Finally there is an area for further research related to a interesting find that implies that chats representing games from competitive ESports categories have more peaks in topic percentage than more casual games.

1 Introduction

Twitch.tv is a website that allows users to broadcast themselves to an online audience performing a range of activities [10], known as "live-streaming". There are several categories, including Music, Food and Drink and Just Chatting, however the category that this paper will be focusing on is video games. Streamers, the people who host their own streams, record themselves and their gameplay, and broadcast this through the site to potentially thousands of viewers. Each livestream is accompanied by a live chat, which allows the audience to communicate with both each other and the streamer. This is presented in the form of a scrolling window displaying recent messages to the streamer and each viewer [5]. Viewers are able to freely send messages in this chat, although they are often moderated to keep out distasteful content [11]. Recently there have been many studies into the nature of the livestream chat and research has shown that many valuable insights can be gained from analysing the content of livestream chat data.

Previous research into the livestream domain has focused on differing topics. Some research has focused on the utility of the livestream chat, for example using it to support highlight extraction [12]. Other research has revealed the event driven nature of the chat, comparing the content of ESports (Electronic Sports) chats to being in a sports bar watching live sport [9]. However an area of research that has been investigated multiple times, is how the chat engagement differs in relation to the size of the audience. [1], [2].

The research question for this report is as follows:

How does chat engagement vary between the audiences of different streamers, and why?

The first step is to define "different streamers". Streamers can be differentiated in a number of ways. Recently, a common approach has been to separate different streamers by their audience size, and this has been researched relatively extensively. However there is a seemingly large gap when it comes to separating streamers by the genre of game they are playing. We know that chat engagement differs by audience size, with larger chats displaying a shift towards "crowdspeak" [1], but there is little information about how the genre of game being streamed affects the chat engagement.

Therefore this report will categorise "different streamers" by the genre of game they are playing. The research will focus on how the genre of the game being played by a streamer, affects the engagement of their audience. More specifically, the difference will be measured between games which can be classified as "shooters" and "non-shooters". These are generalised classes which have been picked because they divide the data set up into 2 parts as close to 50/50 as possible. The information used to select these classes comes from the tags given on the Twitch.tv website. At first the data was divided by the tags "shooter" and "strategy", however this was soon adjusted to a "shooter" and "non-shooter" divide which best captured all of the games in the data set. Games that are classified as "shooter" are those where the primary objective revolves around players shooting their opponents to score well. Games classed as "non shooter" can still involve combat however this won't particularly be focused around gunfights.

To measure "chat engagement", there first needs to be a model created which can process the livestream chat data and give an interpretable output. Therefore, the methodology to investigate the question will be based on the implementation of four topic models, fitted to streams relating to four separate games, two shooters and two non-shooters. The data comes from the TwitchChat data set [5] and the topic model of choice is the Latent Dirichlet Allocation (LDA) algorithm. Topic modelling is a form of statistical modelling that discovers abstract 'topics' in a document. A document has many topics, and each topic is comprised of many words [6]. In the case of the TwitchChat data set, each csv containing chat logs would be considered a document. The choice of LDA was one based purely on the popularity of the algorithm, something that can be seen by a quick google search. These abstract 'topics' will describe the nature of the chat engagement in the streams relating to each of the four selected games. Each topic may represent a different mood or a set of reactions which are typically used side by side. By comparing these topics, we can compare how the audiences of different streamer's engage with their streams, including how the reactions to key events differ.

Chat engagement will be measured through both quantitative and qualitative methods. Both methods will be related to the topic distribution of the selected streams, a measure of how even (or not) the spread of the topics is throughout the stream. This will be measured by calculating the Gini coefficient across the topics, taking inspiration from Bulygin et al [9]. This is a value which ranges from 0 to 1 and is typically used in economic terms to calculate income inequality [8]. Once calculated, a value of 0 indicates perfect income equality, whereas a value of 1 indicates perfect income

inequality. In terms of this project, a value of 0 will indicate perfect equality between the topics in the stream, meaning each topic gets the same amount of "air time" in the chat throughout the stream duration. However, a value close to 1 will indicate that one topic is getting much more "air time" than the other topics.
Quantitatively, chat engagement will be defined by the Gini coefficients of each stream. Comparing these values between streams will highlight potential differences and similarities in audience chat engagement through topic distribution.
Qualitatively, chat engagement will be defined by the nature of the topics learned by each model. These topics will be analysed, specifically the ones which cause peaks in Gini coefficient over the duration of each stream. This will highlight how audience reactions to key events differ between the streams and genres. This is of course under the presumption that sharp increases in chat activity in a specific topic, correlate with stream highlights. To potentially aid both the quantitative and qualitative analysis, the topic percentages will be plotted across the duration of each of the chosen streams. This will show how the prevalence of each topic varies throughout the stream duration, and help find peaks in audience activity that may go unnoticed by the Gini coefficient.

So we have the following subquestions, where "different streamers" will be categorised by the genre of game being played:

- How does the Topic Distribution vary between the audiences of different streamers?
- How do the nature of the topics in the chat vary between the audiences of different streamers?

The report will be broken down into four sections. A Literature Review, A Methodology section, a Results section and a Conclusion

The literature review will focus on the background of the area. Analysing several papers in the Twitch livestream chat domain and their utility towards the project. In the methodology section, the steps taken to implement the model will be described, showing in reproducible terms the procedure that started with raw unprocessed data and ended with four trained topic models. Results and Analysis will focus on the results gathered by passing chat data through the models. The analysis will come in both quantitative and qualitative forms, focusing on the distribution of the topics in certain streams, and the nature of said topics. Finally, the conclusion will summarise the findings and raise areas for future research in the Twitch live stream chat domain.

2 Literature Review

2.1 Introduction

Naturally there are many different approaches that have been taken to investigate this relatively new field and this chapter presents a brief discussion of some of these perspectives. Where possible, papers have been grouped together based on their content.

Briefly, we first have a discussion of the central paper to the project which provided the data set [5]. This is followed by a discussion of the effect that audience scale has on livestream chat engagement from both [1] and [9]. Following this, we introduce the algorithm of choice for the project in the form of topic modelling, including topic distribution, a key metric used to measure chat engagement. Finally there is an overview of an example of the wider background of this topic area, with [12] showing a deep learning model which utilises chat messages to extract stream highlights. We then summarise the findings gathered from the literature and indicate how the model proposed will fill the gap in the current research.

2.2 TwitchChat: A Dataset for Exploring Livestream Chat

The first paper provides the fundamental data set for which the project is based on. With 60 million tokens of live stream chat data, each with matching metadata and spanning a variety of different genres, this paper is essential. There is a detailed description of how the data set was gathered and cleaned up, followed by an investigation into several word vectorization models. Data was gathered from livestreams on 20 different games, with the most popular stream for a set of games at the time being used as the source. The data cleaning process involved a 3 step method. First the cleaning of tokens which involved removing stop words and lemmatization. Secondly, token selection which involved selecting a subset of the tokens to train the model on. Tokens were collected for the subset based on their ‘document frequency’ instead of ‘raw frequency’, which gave priority to the tokens that appeared across several documents rather than lots of times in only a few documents. Last was the final cleaning stage which involved removing all the tokens that were not selected in the previous step. Following this is the case study which compares several different SGNS (skip gram negative sampling) word vector methods, aiming to highlight the unique features of the TwitchChat data set. This includes a traditional SGNS model, A model with dynamic windowing (DW-SGNS) and two temporal SGNS variants, 2T-SGNS and 5T-SGNS. SGNS models “use spatial distance between tokens as cue for semantic similarity”.

The data set and explanation provided by this paper make it essential to the project. The paper acted as an intriguing starting point for the research required and opened up a lot of paths for further reading. Furthermore the code provided alongside the dataset is very useful for working with the data. It has been designed specifically to work optimally with the dataset and will be very useful in the data cleaning process before fitting the data to the topic models. The experiment is an interesting one however it does not relate to the implementation for this solution which will focus on topic modelling using an LDA model.

2.3 Assessing the effect of audience scale on chat engagement

The effect of scale on chat engagement was initially investigated by Ford et al in their paper Chat Speed OP [1], focusing on the popular card game Hearthstone. This research was developed a few years later in 2019 by Flores-Saviaga et al [2], who clustered stream audience sizes into 5 distinct groups.

The two papers differ in their approach to the problem. Ford et al [1] measured 3 defined metrics to analyse chat segments. For each message segment they measured the message length, amount of original content, and the number of unique voices, hypothesising that all three would be smaller in the ‘massive’ chats than the smaller chats. Whilst this was true for the first two hypotheses, the number of unique voices remained consistent throughout. Flores-Saviaga et al [2] took this a step further, and introduced sentiment analysis to try and uncover the emotional feeling in the chats. This is a form of analysis that is used to determine to what degree an expression is positive, negative or neutral. This revealed that the messages in the larger chats were more negative in their tone than the smaller chats. The two papers’ approach also differs by the number of categories they decided to split audience size into. Ford et al focused on the effect of 'massive' chats (more than 10,000 viewers), and included the small chats (less than 2000 viewers) just for comparison. However Flores-Saviaga et al provided categories for a larger spectrum of audience sizes, using a mean shift algorithm to cluster the streams based on view count. The five groups are defined as follows in Table 2.1:

Table 2.1: Stream types and viewer count as defined by Flores-Saviaga et al [2]

Stream Type	Viewer Count
Clique	0-6 viewers
Rising Streamers	6-1879 viewers
Chatterbox	1879-7703 viewers
Spotlight streamers	7703-21678 viewers
Celebrities and tournaments	21678+ viewers

Both papers found significant correlation between audience size and differing chat engagement. For one, the results from both papers indicate that there is a negative correlation between the size of the chat and the length of the messages. In [1] this is defined as "shorthanding", which together with "bricolage" (reusing a small number of emotes and phrases) and "voice taking" (the adoption of shared viewpoints) forms a level of coherence in massive chats which they define as "crowdspeak", a sort of controlled chaos. The results from [2] indicated a positive correlation between the size of the audience and audience retention, with the viewers of larger streams tending to watch for longer. There is also a correlation between the sentiment in the chat and the size of the audience, as the messages in the larger chats are seemingly more negative in their tone.

Both papers provide a useful background for Twitch livestream chat analysis and they both demonstrate that there is a clear correlation between chat engagement and audience size. Ford et al provide a call to further research, which provides clear motivation for the project. In particular, research into "the extent to which the morphology of crowdspeak may be affected by factors such as platform, primary language, streamer, and topical focus". 'Streamer' and 'Topical focus' are both terms that can be investigated through examining the effect that different genres of game have on audience chat engagement. Hence this paper presents a strong motivation towards the research area of the project.

Flores-Saviaga et al provide Sentiment Analysis as an option for investigation which theoretically could be very useful. It would indeed be very interesting to see how the sentiment of audience engagement differs between audiences of different genres. However this task is very difficult on Twitch data due to its abnormal nature [5]. Furthermore the paper lacks clear instructions of how this analysis was performed, despite the difficult nature of the corpus. Unfortunately this rules out sentiment analysis as a viable option, but regardless of this the background and context provided are still invaluable.

2.4 Using Topic Models to analyse Twitch Livestream Chat data

Here we discuss a paper and a blog post which both implement forms of topic modelling on Twitch livestream chat data. Naturally, the two pieces of literature differ in their aims and objectives. With [9], Bulygin et al implement a structural topic model on livestream chat data from different stages of a Dota 2 tournament. They compute the Gini coefficient on the topics for these streams and compare this value between the different stages of the tournament, showing for example, that the topics were much less evenly distributed in the finals than the group stage. With [7] Bhatia provides a relatively straightforward python implementation of an LDA topic model on Twitch chat data. This covers collecting the data, preprocessing, fitting the models and some analysis. The tone is very different to [9], resembling a tutorial more than an academic research paper.

2.4.1 Between an Arena and a Sports Bar: Online Chats of Esports Spectators

Between an Arena and a Sports Bar [9], provides an example of Topic Modelling used on documents gathered from the Twitch livestream chat. More specifically, the data was gathered from the chat of several stages of a Dota 2 tournament. Here Bulygin et al use Topic Modelling to investigate topical and temporal patterns of chat messages and investigate their relation to in game events. They show that "in-game events drive the communication in the massive chat and define its emergent topical structure to a various extent". That is, the topics in the chat are directly related to the emergence of in game events. Due to the source of the chat data, a direct comparison between the different stages of the tournament was possible. There are two key introductions given by this paper that shall be discussed, the introduction of Structural Topic Modelling (STM) and the introduction of the Gini coefficient to measure the topic distribution in the live stream chat.

The first key aspect of this paper is the introduction of a specialised form of topic modelling, Structural Topic Modelling (STM) [13]. This model allows the input of document level metadata to influence the learning of the topics. By doing this, the model addresses some of the limitations of LDA, most notably the fixed distribution of words per topic and the lack of document contextual information considered when deciding topics. STM allows different documents to talk about the same topic with different words [14]. For example, topics of conversation where opinion on the matter would make the words used to describe the events differ. So STM allows for a more flexible and potentially more powerful approach to topic modelling than LDA. In the case of this paper, this metadata was the current tournament stage that was being broadcast, which was used to influence the learning of topics by the model. At first glance STM would seem ideal for the project at hand: The TwitchChat [5] data set contains a variety of metadata including user view count, user broadcaster type, stream game id, and stream viewer count. However STM is currently a model which is only available in R. As Python is the preferred programming language for this project, unfortunately this model cannot be used.

Despite this there is still much of use to be learned from this paper. The introduction of the Gini coefficient for Topic Distribution is key and potentially provides a useful form of measurement in the project. The Gini coefficient, once calculated, is a value that ranges from 0 to 1, "0 means absolute equality among topics... and 1 means that only one topic is present while others are missing" [9]. In the paper, Bulygin et al compare the coefficient between different tournament stages. This implies there is potential to calculate and compare this coefficient between several LDA models, each trained on different games representing the shooter and non-shooter genres. This could clearly indicate whether or not the genre of game being played has a significant impact on the topic distribution in the respective livestream chat. Therefore this will be a key metric to help answer the major question of how chat engagement varies between the audiences of different streamers. Perhaps distribution of topics is broader in one genre over the other, implying the existence of more voices in the chat [9].

2.4.2 Natural Language Processing for Post Livestream Analysis

Bhatia provides a blog which demonstrates a step by step guide for implementing Latent Dirichlet Allocation in Python for use in analysing livestream chats [7]. Alongside the guide is a link to a git repository containing the code that he uses. Firstly there is a clear method for gathering and putting Twitch chat data into a pandas data frame. Next there is a guide on preprocessing the data, making use of the following techniques:

1. Tokenisation (using the Genism library)

2. Removing stopwords (using the textHero library)

3. Lemmatization (using the Spacy library)

4. Stemming

Bhatia provides definitions for each of these techniques. Firstly, tokenisation is described as the action of "Splitting the text into sentences and those sentences into words". In essence, this splits up each sentence into different 'tokens' that can be put into the LDA model. Secondly, stopwords are said to be "common words that would appear to be of little value". These are the words that appear so often, they hold little meaning and they would be of little value when splitting a document up into discrete topics. Because of this, these words are removed. Next we have lemmatization, "changing words to a first person form and to the present tense". The reason this is used is so the model can group words that mean the same thing together. For example "run" and "ran" both describe the same action, just in a different tense. Therefore after lemmatization, "ran" would become "run". Finally, we have stemming. This is the act of "reduc[ing] [words] to their root or stem form". For example, the words "waiting", "waited" and "waits" would all reduce to the same stem form, "wait" [15].

Next Bhatia describes the method to create and visualise the LDA model. The LDA model is created using the commonly used machine learning package scikit-learn. To visualise the different topics across the document, the pyLDAvis package is used. The use of pyLDAvis allows the creation of a "intertopic distance map" which is a "visualisation of the topics in a 2D space". In this map, the distance between the topics is significant, as the closer two or more topics are together, the more words they have in common. Finally Bhatia shows how to implement Document Search making use of Semantic Text Similarity. This allows the corpus to be searched by topic and to see what people were saying about that topic.

The combined utility of these two papers is incredibly useful for the project. The structure given for the LDA implementation in [7] provides strong foundations for the implementation of the LDA model with the necessary modifications being simple to implement. Of these modifications, most notably there will be several models trained rather than just one, the method for removing stopwords will be altered to better fit the task at hand, and a more optimised form of lemmatization created specifically for the TwitchChat [5] data set will be used. Furthermore the searching techniques and the intertopic distance maps are not entirely relevant to the project so these will be omitted. The Gini coefficient used in [9] provides an excellent tool for analysing the chat engagement once the topic models have been trained. This will be key for both quantitative and qualitative analysis. The topic distribution is an interesting metric on its own and but it also opens up the opportunity to discover and analyse the nature of the topics that cause peaks in chat interest (see 4.2). Furthermore, regardless of STM being inaccessible, [9] shows that valuable insights can be indeed be retrieved from running a topic model on Twitch livestream chat. This motivates the decision to use topic modelling as the form of machine learning for the project.

2.5 A Deep Learning Model for Extracting Live Streaming Video Highlights using Audience messages

With this paper, Han et al provide a new method of extracting highlights from livestreams. Rather than the traditional method of analysing video frames, they propose a deep learning model which analyses the live stream chat instead [12]. To embed the data, Han et al make use of the word2vec embedding function, then they pass the tokens through their recurrent neural network. The use of a recurrent neural network is important in Natural Language Processing. This is because they are effective in "retaining contextual information when processing sequential data". The model proposed consists of two networks, "a two layer bi-directional GRU network and a multiple-layer deep neural network" (DNN). Here the DNN is key, it analyses audience responses to see where a highlight has occured. The DNN does this by learning the semantics of messages that are associated with highlights. The model proposed, biGRU-DNN, is very successful. It outperforms several baseline methods which work by analysing video frames.

This is an interesting paper and provides worthwhile context to the subject area. However, it isn't as directly useful to the project as the other papers. This project will focus on a topic modelling approach to extract information about the chat, rather than using the chat to extract highlights from the stream. Therefore, this paper provides an interesting background of the wider research area, but is not specifically related to the project.

2.6 Summary

To summarise, the papers each bring different angles of utility to the project. With the first paper [5], we are provided with the TwitchChat data set which is the centre of the research throughout the project. [1] and [2] highlight a gap in the research area, which is the impact that genre has as a variable in differing chat engagement. While [9] and [7] provide the foundations for the means to create the necessary model to fill this gap. From both [1] and [2], there is an interesting discussion about the effect of scale on audience participation in livestream chats. This provides background into the research area, displaying that scale has been considered to be an important variable in differing chat engagement. An appropriate model and foundations for analysis are provided by [9] and [7]. What is particularly useful is the introduction of topic distribution as a metric for measuring chat engagement. Using this metric opens the opportunity to look not only at the nature of the topic content in the different streams, but also the nature in which these topics appear. This paves way for an understanding in which topics are more frequent than others, and which chats are more or less diverse in their topic content.

3 Methodology and Design

3.1 Introduction

The methodology for the project centred around fitting a topic model to livestream chat data to try and uncover valuable insights. Topic modelling was chosen due to inspiration from the implementation used by Bulygin et al [9]. Extracting the topics in the livestream chat seemed like the perfect way to begin to measure chat engagement. The first step was to collect and prepare the data. Luckily, collecting the data was made simple as this had already been done [5]. Next the data had to be imported into python and processed in preparation for an LDA model. An important decision had to be made here. Either the data could be split into the two groups shooter and non shooter, or two games from each genre could be picked.

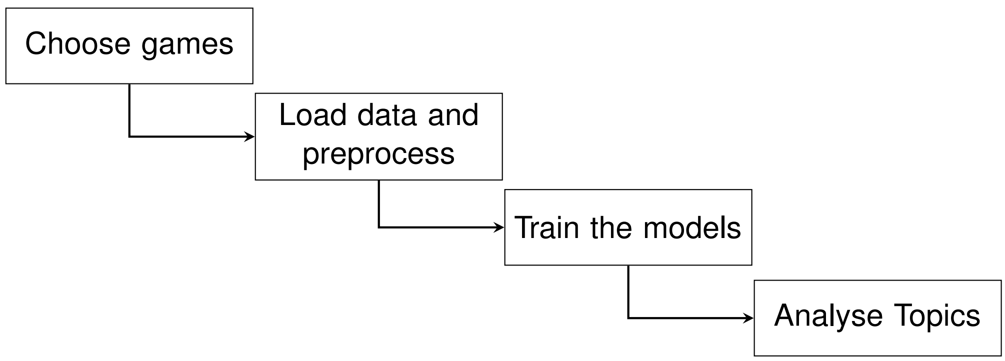

Figure 3.1: Methodology pipeline

3.1.1 Option 1: Splitting the dataset in half

Initially the data set would be split (roughly) in half by genre. Theoretically this made the most sense: the aim of the model is to compare the topics of shooters and non-shooters, so it stands to reason that two models should be trained on data from each of the two genres. However in practice this presented two problems. Firstly, in order to reduce training time, the models would have to be fitted using a subset of the data. Choosing a subset of 1 million tokens provided a reasonable training time, with any increase on this creating an unreasonable wait, especially as this is an iterative process involving parameter tuning. However, there were 6 million tokens for the shooter data and 8.5 million tokens for the non shooter data. A subset of 1 million tokens therefore would miss out a significant portion of each data set. Furthermore there would be little information on where the selected tokens came from, and how many games were present. Secondly, with only one one topic model for each genre, it wouldn't be possible to look within the genres for insights, only between each genre. Insights about differences are hard to establish if there isn't a null hypothesis for comparison, that it cannot be shown that these differences don't happen between games of the same genre. Therefore another solution was needed.

3.1.2 Option 2: Choosing four games

The solution to the above problems was rather simple. Instead of dividing the data set in two, a selection of four games would be chosen. Within these four games, two would be from the shooter category and two from the non-shooter category. Immediately this would allow for insights to be gathered within genres as well as between them. As, for example, shooter-shooter comparisons could be made. Furthermore the issues causes by sub-setting the data could be mitigated. A subset of 500,000 tokens was chosen, which is a lot less damaging on a data set of just over 1 million tokens, or 3 million at most, than a data set of 8.5 million tokens. Furthermore all of these tokens would be from the same game, so there would hopefully be a limited amount of context lost when using the subset. Moreover, this subset gave a fast total training time of just under 30 minutes.

Therefore four separate groups were selected from the data set, including two games which could be classified as shooter, as well as two non-shooter games. The games were chosen based on their popularity, which was decided based on the total view count for the streams corresponding to each game. As can be seen in Table 3.1, the two most popular shooters were Fortnite and CounterStrike: Global Offensive, with League of Legends and World of Warcraft being the two most popular non-shooters. Following preprocessing, four separate LDA topic models were fitted to the subsets of each of the chosen game's chat data. Once trained, the models could be compared and differences and similarities highlighted.

3.1.3 A Brief description of the chosen games

Fortnite is a battle royale style multiplayer third person shooter. In this game, 100 players are dropped onto a large island. Here they must find better

Table 3.1: The top five most viewed games in the data set

GameID	Title	Shooter	Total View Count
33214	Fortnite	True	2325669
18122	World of Warcraft	False	2021683
21779	League of Legends	False	1794139
32399	CounterStrike: Global Offensive	True	1025786
138585	Hearthstone	False	774224

equipment and fight other players until they are the last ones standing [16].
CounterStrike: Global Offensive (CSGO) is a multiplayer first person shooter. The game revolves around objective based modes where two teams, terrorists and counter-terrorists, go head to head [17].
League of Legends (League, LoL) is a multiplayer online battle arena (MOBA) game. 2 teams of 5 players go head to head to try and win control of their enemies section of the arena [18].
World of Warcraft (Warcraft, WoW) is a massively multiplayer online role playing game (MMORPG). Players create a character and explore an expansive fantasy open world, completing quests and levelling up their character [19].

3.2 Collecting and preparing data

3.2.1 Loading the data

The TwitchChat dataset contained a plethora of chat logs, each stored in csv files and accompanied by matching meta data stored as json files. In order to work with both sets of files, they would need to be loaded into python. To do this, extensive use was made of the python library pandas [20]. This commonly used python package allows the creation and manipulation of data frames. Firstly, the json metadata files were loaded into a dataframe, keeping the headings consistent with those in the files, and adding a new heading to store each entry's file name. Next a game-genre-views lookup table was created, making use of the GameID -> Game Title translations that were provided. This would be necessary for the next step which was to create four dataframes, to store the chat logs of each of the four chosen games. In order to decide which games to use, their total viewing numbers, provided in the metadata, were summed up. This gave the two most popular shooters, Fortnite and CounterStrike Global Offensive, and the two most popular non shooters, League of Legends and World of Warcraft. All of the csv files were imported into a list. For each csv in the list, a pandas dataframe was created, storing the chat log. This was then added to one of four lists, depending on the game. Finally the dataframes in each list were concatenated together, leaving four dataframes, each containing chat logs across streams of each game.

3.2.2 Preprocessing for LDA

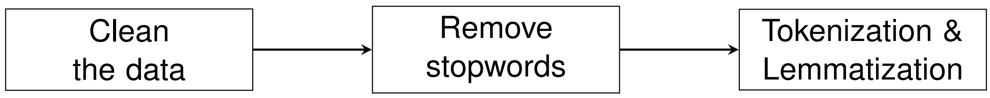

Figure 3.2: Preprocessing pipeline

Now the data was loaded into dataframes, pre-processing for LDA could begin. This largely followed the steps presented by Bhatia [7], however the lemmatisation code came from the TwitchChat paper as it was specifically created for the data set. The steps are highlighted in Figure 3.2. Bhatia's implementation made extensive use of the Texthero library, which provides a pipeline to clean and represent data [21]. The first step in data cleaning involved setting each chat entry to lowercase, removing punctuation and removing URLs. Next was the removal of stopwords. Here the implementation differs from that given by Bhatia. His implementation removes custom stopwords based on the common phrases in the Twitch chat, however in this case this would've resulted in a significant loss of information. Removing stop words such as emotes would remove vast amounts of important data from which insights can be gathered. By analysing emotes, the feeling in the chat could potentially be recovered, so it was vital that these were not removed. The next step was tokenization, which involved splitting sentences up into words, making use of the Genism topic modelling library [22]. Finally lemmatisation was performed using the TwitchChat data set code. This code used the standard Natural Language Toolkit (NLTK) [23] lemmatisation code as well as custom lemmatisation techniques based on Twitch specific words (such as emotes and reactions like "haha").

3.3 The Model

3.3.1 Training the Model

To model the data, topic modelling was performed using Latent Dirichlet Allocation, available in the scikit-learn package [24]. Before the lemmatised data could be modelled, it first had to be vectorized. To do this, the

CountVectorizer function was used, again from scikit-learn. Both the minimum number of occurrences required for a word to be considered and the number of characters per word were set at their default values of 10 and 3 respectively. The next step was to fit the vectorized data to each respective LDA model. After several attempts, the best number of topics for the data was set at 5, any number higher than this made the topics increasingly difficult to interpret. The models were fitted with a subset of 500,000 tokens for each of the games, roughly 1/2 of the Fortnite tokens, 1/3 of the CounterStrike and League of Legends tokens and 1/5 of the World of Warcraft tokens. Once fitted the top 10 words for each topic were put into 4 dataframes which allowed for a visualisation of the content of each topic.

3.3.2 Quantitative Analysis: Topic Distribution

To analyse the data, a mixture of qualitative and quantitative methods were used. Quantitatively, calculation of the Gini coefficient enabled a measure of the topic distribution across each game. The Gini coefficient is a value in the range 0-1 that represents topic distribution in the following way: a value of 0 indicates total equality amongst topics and a value of 1 indicates complete inequality [9]. In order to calculate this, a stream from each game had to be fitted to its corresponding topic model, with the streams being chosen by the highest number of views. Once the stream messages had been fitted to the model, the topics were calculated for 5 second windows, starting from 0-5 seconds, 5-10 seconds etc. This allowed for a visualisation of the topic prevalence at each stage in the stream. To calculate the Gini coefficient, the topics were categorized, with only the most abundant topic for each window kept. This gave a max topic list showing the most common topic for each 5 second window in the stream. On this data the Gini coefficient could be calculated, showing the distribution of the max topics. The Gini coefficient was calculated with a moving window across the max topic list. By calculating the coefficient on many sub-lists, it could be plotted over time to show how the topic distribution changed throughout the stream, rather than having just one value to summarise the entire stream. This distribution could then be compared for each game and genre. For readability, the plots were smoothed using the Savitzky-Golay filter, a function available in the SciPy package [25]. The length of the selected streams was not consistent, as view count was the only factor considered for selection. Because of this, the Ginis had to be plotted over the course of the shortest stream, which was the stream chosen for League of Legends.

3.3.3 Qualitative Analysis: Topic Nature

Finding meanings for topics was a qualitative task. The method involved reading the top words for each topic, usually emotes, and trying to construct meaning from these sets of words. Perhaps looking for topics such as celebration, laughter or boredom. Then the nature of these topics could be compared across the games and across the genres these games represented, to try and find differences in audience chat engagement. To apply this analysis, this section built on the work of the previous section, looking deeper at the peaks in Gini coefficient for each stream. Finding out the nature of the topics that caused these peaks, would lead to a good indication of how each chat engaged with their respective streams, showing which topics sparked the most interest for each audience. The most significant peak was chosen for each stream, and the most prevalent topic in this window was recorded. Next, qualitative analysis on these topics was applied to try and understand what they represented, and this was compared for each stream.

3.3.4 Topic Percentages

To support both quantitative and qualitative analysis, the topic percentages were plotted over the duration of the chosen streams. This is a value that indicates how much a topic is represented in a given window of messages. The streams were once again fitted to their corresponding topic models, and the topics recorded for each 5 second window. However this time, instead of categorising the topics by the max topic, all 5 topics were kept. By plotting the topic percentages, the way that each topic varied in prevalence throughout the duration of the streams could be visualised. Comparing the shape of the plots would extend the quantitative analysis and the nature of the peaks in certain topics could be analysed qualitatively. To make fair comparisons between the shape of each plot, the length of the streams would need to be similar, and the range of the y-axis would need to be consistent. Normalising the y-axis was a simple fix, but two new streams would need to be found to make the length consistent. A new, longer, League of Legends stream was selected and a shorter CounterStrike stream was selected. Then only the first roughly 25000 messages were plotted for each game, in order to keep each plot the same size as the smallest stream (which was Fortnite).

4 Results and Analysis

4.1 Topic Distribution with Gini coefficient

Over the duration of each livestream, the Gini coefficient varies over a range of values, as displayed in Figure 4.1. Smoothing has been applied here as it has in the rest of the plots that follow. The importance of applying smoothing for legibility can be seen by comparing the plot in Figure 4.1 to the not smoothed version in Figure A.6 located in the Appendix.

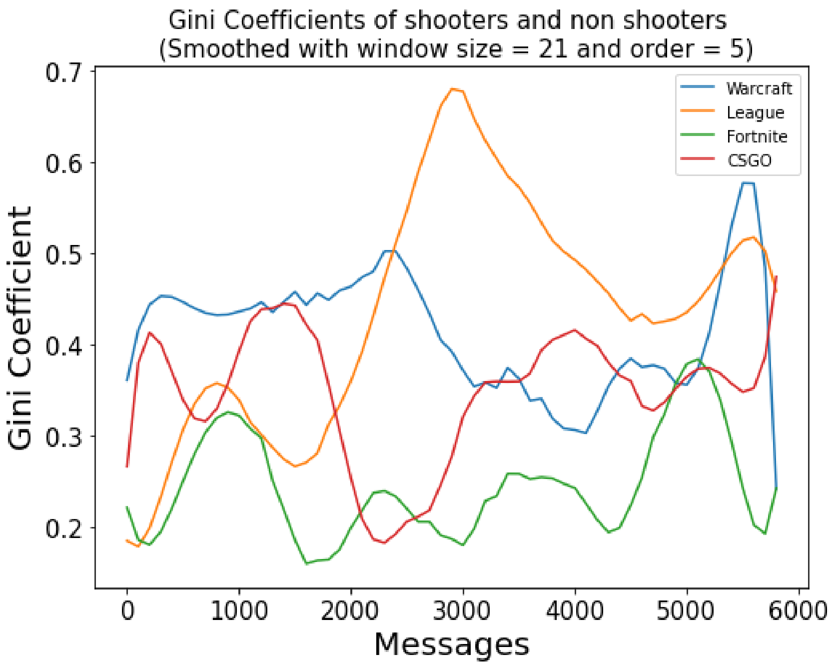

Figure 4.1: Gini coefficient across all four games. Smoothing has been applied using the Savitzky–Golay filter

By looking at Figure 4.1 we can see that the Gini coefficient in the League of Legends stream varies from 0.2 at the beginning to a peak of 0.7 about halfway through. At this peak, one topic was dominating every other topic, almost becoming the sole voice in the chat. It is likely that this spike coincides with a key moment in the stream, which can perhaps be figured out by looking at the specific topic that peaked (see section 4.2). We can also observe that at several points, both non-shooter games have a higher Gini coefficient than the shooters. From about 2000 messages onwards, the League of Legends Gini dominates both Fortnite and CSGO. The Warcraft Gini dominates both Fortnite and CSGO up to around 3000 messages and then starts dominating again just after the 4200 mark. This of course would indicate that the topics in the non-shooters are less varied than their shooter counterparts. However if we consider the median and interquartile range's of the Gini coefficients across the whole streams, we see that these are a lot closer in value than indicated by the graph.

Table 4.1: Gini coefficient Statistics

Game	Median	IQR
Fortnite	0.228	0.128
CounterStrike: Global Offensive	0.380	0.164
League of Legends	0.452	0.228
World of Warcraft	0.388	0.148

Table 4.2: Shaprio-Wilk Test for Normality. If p-value < 0.05 then the distribution is not normally distributed

Game	Test Statistic	p-value
Fortnite	0.940399	0.000000
CounterStrike: Global Offensive	0.971615	0.000000
League of Legends	0.968231	0.125604
World of Warcraft	0.996965	0.050054

After applying a Shapiro-Wilk test for normality (as shown in Table 4.2), only the Gini coefficients from the League of Legends stream were confirmed to be normally distributed, with a p-value > 0.05. Because of this, it was favoured to treat the data as non-parametric, and examine the median and interquartile range of the Gini coefficients, rather than the mean and standard deviation. In Table 4.1 and we can see right away that, as indicated by Figure 4.1, Fortnite has the lowest median Gini coefficient and League of Legends has the highest, with 0.228 and 0.452 respectively. Furthermore when consulting the interquartile range, we can see that Fortnite, with the lowest IQR, has the most consistent topic distribution throughout the stream, whilst League of Legends, with the highest IQR, is the least consistent. However, despite being from different genres, CounterStrike Global Offensive and World of Warcraft have a very similar topic distribution, with their median Gini coefficient's only varying by 0.008. This is further emphaised by the p-value given by the Mann-Whitney U test between the two games in Table 4.3 which, at 0.41, is far above the threshold of 0.05. In fact, these are the only two games where the distribution of Gini coefficients is not significantly different.

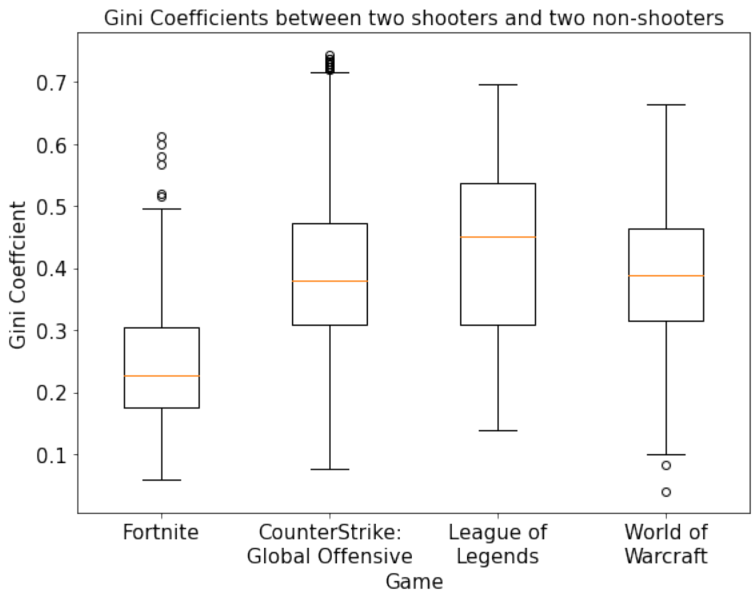

Figure 4.2: Gini coefficient across all four games

The box plot provided in Figure 4.2 helps visualise the topic distributions more clearly. The large interquartile range for the League of Legends stream can be seen by its large box, as can the small IQR for the Fortnite stream. We can also see that the Fortnite topic distribution is significantly more even than the other games, with its plot being much lower down the y-axis. This plot further emphasises the gap between the non-shooters on the right and the shooters on the left as they dominate by their median average Gini coefficient.

Table 4.3: Mann-Whitney U test on the Gini coefficients from each game. If p-value < 0.05 then the distributions are significantly different

Game 1	Game 2	Test Statistic	p-value
Fortnite	CounterStrike: GO	111314.00	0.00
Fortnite	League of Legends	2392.00	0.00
Fortnite	World of Warcraft	48078.00	0.00
CounterStrike: GO	League of Legends	57589.50	0.019089
CounterStrike: GO	World of Warcraft	1172072.00	0.415320
League of Legends	World of Warcraft	24324.50	0.007427

So we have a case where there is a clear difference in Gini coefficients between the genres (Fortnite and League of Legends), and a case where this difference is a lot less significant (CounterStrike and World of Warcraft). Therefore the results show somewhat of a difference in topic distribution between the genres, with the non-shooters displaying a slightly less equal topic distribution, indicating a slightly less even prevalence for each topic.

However, we must consider the length of the stream window reported in Figure 4.1. Although this represents the whole League of Legends stream, it only contains a small subset of the other 3 streams, each containing many more messages after the 6000 mark. Therefore these other streams could have different variations in Gini coefficient later on in their duration which are not accounted for by this plot. However these are indeed accounted for by the statistics in Table 4.1, which were calculated for the entire duration of each stream.

4.2 Qualitative Analysis of Topic Nature

4.2.1 League of Legends

To begin qualitative analysis, we start by looking at the Gini coefficient spike in Figure 4.1. The high value of 0.7 indicates that one topic was dominant at this point in the stream. The topic in question, was discovered to be Topic 1. In Table 4.4 we can see the top words for each topic, and using this information we can try to derive a meaning for Topic 1. The words "haha", "kekw" and "lol" are all responses to humorous events in the stream [26], as is "pepe" depending on its suffix (which we can presume to be "laugh" due to the other words in the topic, creating the emote "pepelaugh" [27]). Next is "monkaw", which is often used in a response to tense situation [28]. The emote "trihard" has a few uses, it can be used to indicate someone is "trying too hard", or also in relation to "fear, surprise or ... nerve wrecking moments" [29]. We can ultimately presume that the peak in this topic highlights an exciting, tense and perhaps humorous part of the stream, certainly a moment of importance. For example this could be an intense battle, a heart in mouth moment or perhaps the streamer winning a game.

4.2.2 Fortnite

When looking at the Gini Coeffcients of the whole Fortnite stream, there is a similar spike about halfway through the stream (see Figure 4.3). Again, this was found to be Topic 1. Another peak follows this around 17000 messages in, however this will be discussed later in Section 4.3.

Table 4.4: Top 10 Words per Topic for League of Legends

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
pepe	lul	lulw	pog	monkahmm
song	boom	clap	way	head
haha	residentsleeper	ratirlmangos	world	quiet
kekw	cmonbruh	kreygasm	wheelchair	let
monkaw	kappa	kong	nam	weirdchamp
trihard	chat	hong	symbol	datsheffy
lol	free	team	monkas	xqcp
bye	game	jebait	wutface	oce
tsm	win	play	heyguys	help
biblethump	bad	good	forsene	copy

The nature of the first peak differs from the one found in the League of Legends stream. Looking at Table 4.5 we can see that both topics share the common word "pepe", however its use may vary. Looking at the other words in Fortnite's Topic 1 we see "kappa" and "residentsleeper". The use of "kappa" indicates irony and sarcasm [30], and "residentsleeper" indicates a late night stream or more commonly, boredom [31]. The use of these emotes together could indicate an ironic feeling of boredom, the content in the game may be slow, or the audience may be taunting the streamer. We also see two streamer specific emotes "ninjagold" and "ninjablast" which, by looking at the emotes, seemingly indicate the streamer, presumably Ninja, performing well [32].

Table 4.5: Top 10 words per topic for Fortnite

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
lul	ninja	pog	love	lol
pepe	trihard	way	text	haha
like	tfuewaaa	biblethump	sym	face
kappa	chat	cmonbruh	tfue	brookeeee
residentsleeper	game	dbe	brooke	tear
ninjagold	tfuesweater	cecd	sub	joy
ninjablast	stream	good	fca	skin
time	chap	lulw	brook	dont
wtf	respond	guy	cloak	ninjaaim
f–k	got	bro	symffist	need

So this topic seemingly implies strong streamer performance and/or taunting. The streamer specific emotes could imply that the streamer is performing well at this moment in the stream, perhaps getting a series of impressive "kills". The ironic boredom implied from "kappa" and "resid-

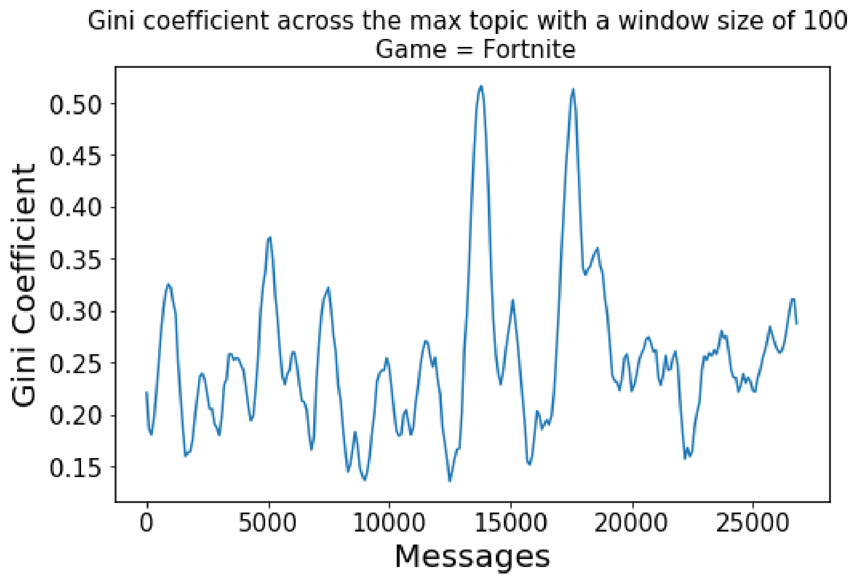

Figure 4.3: Gini coefficient across the Fortnite stream. Smoothing has been applied using the Savitzky–Golay filter

entsleeper" could imply the audience is taunting the streamer. However if the streamer is performing well, perhaps the audience is taunting the streamer's opponent/s instead.

Both of these Topics seem to imply the existence of a highlight in the stream. Perhaps the streamer is performing well, winning a battle or a game. For the League of Legends stream, this could've been a tense situation, followed by laughter which could be aimed at the streamer's opponent. There is a similar theme in the Topic 1 for Fortnite, the streamer specific emotes indicating a good performance from the streamer, and the ironic boredom from the "kappa" and "residentsleeper" emotes could be taunting aimed at the opponent. This shows a similarity between the audience engagement of both streams, however there is perhaps a difference in the way they taunt the opponent. Those from the non-shooter stream favour laughing at the opponent ("haha", "lol", "kekw"), whereas those from the shooter stream prefer being sarcastic and dismissive of their ability, pretending they are tired or bored from their performance ("kappa", "residentsleeper").

4.2.3 CounterStrike and World of Warcraft - Topic Similarities within Genres

Topic 1 also caused the largest peak in Gini coefficient for both the CounterStrike stream and the World of Warcraft stream. Looking at the Topics for both games we can begin to see similarities within the genres.

Table 4.6: Top 10 words per topic for CounterStrike: Global Offensive

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
lulw	pog	haha	pepe	lul
quiet	trihard	biblethump	cache	head
residentsleeper	liquid	new	kappa	silent
telephone	wutface	clap	sweat	steam
receiver	let	kreygasm	droplet	like
squid	way	game	right	major
astralis	god	time	pointing	drinkpurple
map	face	notlikethis	index	start
vac	crowd	xqcp	backhand	dansgame
chat	monkaw	bot	major	rip

Table 4.7: Top 10 words per topic for World of Warcraft

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
pog	lul	pepe	faead	clap
lulw	kappa	jebait	gachibass	kapp
lol	wow	asmonsmash	smorc	true
monkaw	classic	asmon	asmonpls	dansgame
head	good	chat	like	kreygasm
haha	got	game	asmonkool	pjsalt
yes	horde	face	hair	need
trihard	time	f–k	bad	beta
wtf	alliance	music	people	bfa
dont	man	play	f—ing	use

In Table 4.6 for CounterStrike, we can see that Topic 1 contains "residentsleeper" and "lulw", similarly to Topic 1 in the other shooter, Fortnite (although here, "lulw" is replaced by "lul"). Our second non-shooter, World of Warcraft, also shares similarities with its non-shooter counterpart, League of Legends. Table 4.7 indicates that both of games have a Topic 1 which contains the words "monkaw", "trihard", "haha" and "lol". This is quite an interesting discovery as it implies there are consistencies in topics and therefore in chat engagement within similar genres, indicating a correlation between chat engagement and genre of game. Of course, League of Legends, a MOBA, and World of Warcraft, an MMO, are not exactly the same genre, but in terms of neither of them being shooters, they are linked. The link between Topic 1 of the shooters is not quite as strong, with only 2 of the top 10 words being shared, however it still exists. Topics of games from the same genre are similar, but they differ between genres. This implies that topics in the livestream chat are indeed genre dependent, with audience engagement differing depending on whether the game being played is a shooter or not.

4.3 Further insights from plotting topic percentages

Another way of visualising the nature of the topic distribution, is by plotting the percentage of topics over the duration of a stream. This value indicates how prevalent a topic was for a given message in the chat. By plotting these values we can highlight and qualitatively analyse peaks in certain topics that may not be found by the Gini coefficient, in addition to seeing the shape of the topic distributions. Applying smoothing was perhaps more important here than any of the other plots. Without smoothing, the plots were illegible with far too much data being plotted with too little space. For example, see the Fortnite Topic Percentages plot without smoothing at Figure A.7.

Looking at the shooter streams in Figures 4.4 and 4.5, the topic percentage is fairly consistent, with the Fortnite stream having one peak in percentage and CounterStrike having 2 or 3. We can reasonably assume that this means that chat engagement is fairly stable, with only a few spikes.

In the CounterStrike stream (shown in Figure 4.4), there are a few peaks caused by Topic 4. From Table 4.6 we can see this includes the words "sweat" and "droplet". Describing a player as "sweaty" is a common phrase in the gaming community which is used to taunt a player for trying too hard, causing them to "sweat" [33]. It is clear then that "droplet", an emoji depicting a water droplet, would be used side by side with this term, and this could be emphasised by the word "major" also included in this topic (as in ‘major sweat’). At these peaks the chat could either be taunting the streamer, saying they are trying too hard, or perhaps aiming the phrase at the streamer’s opponents. This taunting is emphasised by the inclusion of the previously discussed "kappa" emote which is used to convey sarcasm or irony [30].

In the Fortnite stream (shown in Figure 4.5), there is one noticeable peak in the second half of the plot, caused by Topic 2. Referring back to Table 4.5 we see the words "trihard", "tfuewaaa" and "tfuesweater". The "trihard" emote has been discussed previously, it is often used when a player is "trying too hard" [29], similarly to the word "sweaty" discussed in the previous paragraph. If we assume that the "tfuesweater" emote (a cartoon character wearing a sweater) is also used to imply a player is "sweaty", then this Topic 2 is similar in nature to Topic 4 from the CounterStrike stream. What is interesting about this observation is that both of the visible sets of peaks in the shooter streams are caused by the chat calling out streamers, or opponents for being "sweaty". Therefore, perhaps this culture is related more to shooter games than non-shooters.

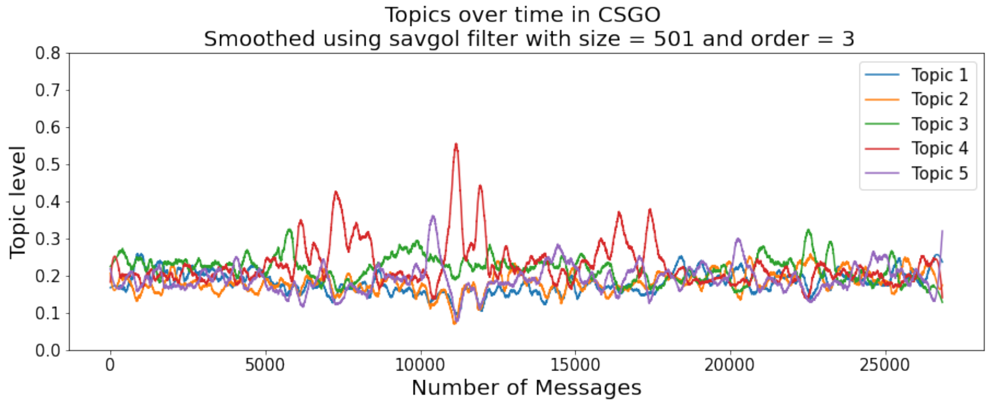

Figure 4.4: Topics over time (represented by number of messages) in a CounterStrike stream. Gini coefficient = 0.34

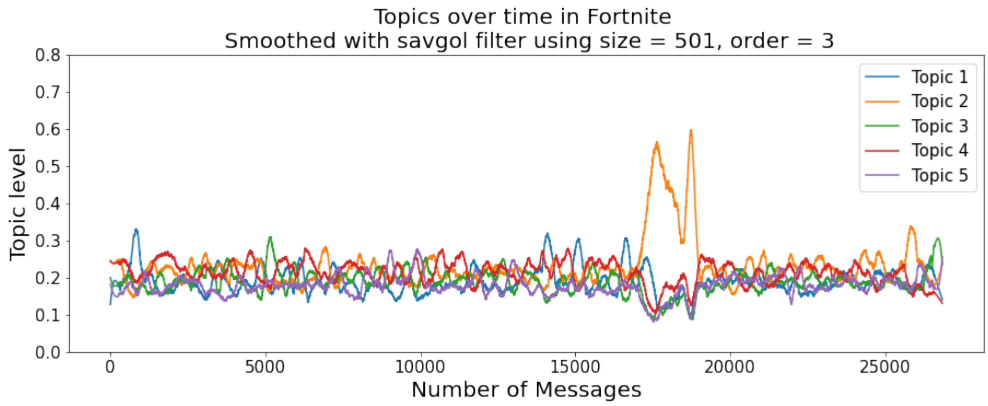

Figure 4.5: Topics over time (represented by number of messages) in a Fortnite stream. Gini Coeffcient = 0.25

The League of Legends stream (shown in Figure 4.6) has a peak much larger than in any of the other streams, with Topic 5 sharply rising and then falling. With all conversation briefly focused on one topic, it stands to reason that this could be a highlight in the stream. Topic 5 includes the emotes "monkahmm", "head" (likely from "4head") and "weirdchamp". Firstly, the emote "monkahmm" tends to be used when the streamer is in deep thought and trying to decide their next move [34]. "4head" is often used in reaction to cheesy jokes or people giving advice that is deemed obvious and therefore unhelpful [35]. And finally, "weirdchamp" is used to express disappointment or disbelief towards the streamer [36]. The word "quiet" is also high up in this topic, perhaps the streamer was trying to plot ("monkahmm") a stealthy attack ("quiet") which quickly backfired ("weirdchamp", "4head"). This would make sense as this peak is shortly followed by smaller peaks in

Topic 2, which includes the previously discussed terms "residentsleeper" and "kappa" which imply taunting, as well as "lul" which is used to express laughter. It would make sense for the audience to laugh and taunt after a streamer's plan backfired.

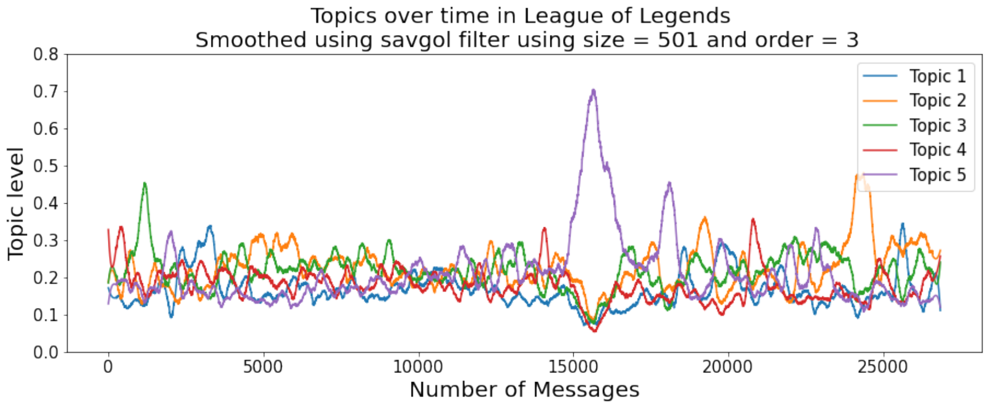

Figure 4.6: Topics over time (represented by number of messages) in a League of Legends stream. Gini coefficient = 0.30

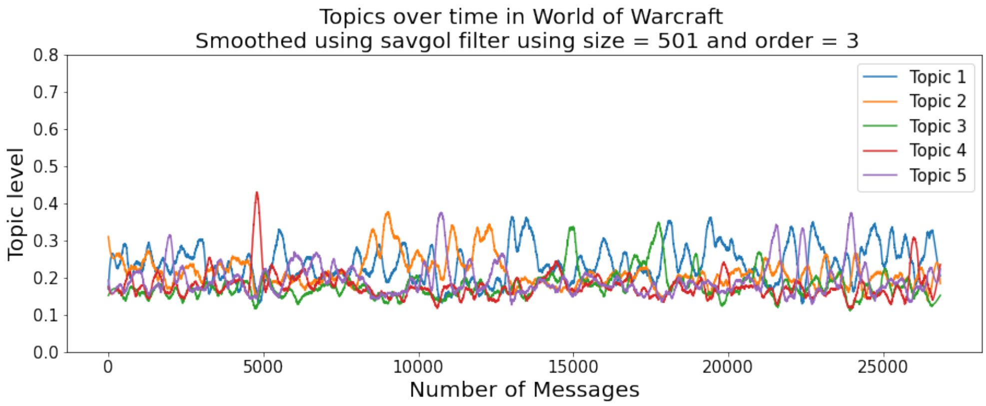

Figure 4.7: Topics over time (represented by number of messages) in a World of Warcraft stream. Gini coefficient = 0.39

The World of Warcraft stream (shown in Figure 4.7) is different from the others because there is no clear peak in topic percentage, with the chat being pretty balanced throughout the stream. It is certainly interesting then, that this stream has the highest Gini coefficient of the four, with 0.39. This could be down to how the coefficient was calculated, with only the max topic being considered. Topic 1 is the max topic quite often throughout the stream, which increases the Gini coefficient despite a lack of any clear spikes. The lack of peaks is interesting and could reasonably be interpreted as a calmer chat, with the engagement being fairly consistent throughout. Possible reasons for this vary, It could be the case that this stream was fairly uneventful. On the contrary, perhaps viewers of this particular game are more casual than the other, more ESports oriented games being played in the other streams.

4.3.1 Inconsistencies between the Gini coefficient and the Topic Percentages

An interesting observation can be made by comparing both the Gini coefficient plots of each stream, and the Topic Percentage plots. As briefly mentioned in the previous section, we can see there are inconsistencies between the peaks in each plot. One would assume that peaks in the Gini coefficient (indicating high topic distribution inequality) would line up with peaks in the Topic Percentage (indicating a high proportion of a certain topic). However, this isn't always the case. Take the Fortnite stream for example. In Figure 4.3 we can see two clear spikes in Gini coefficient, around half way through the stream at roughly 14000 messages in and again at around 17000 messages in. However, if we look at the 14000 message mark in the topic percentage plot in Figure 4.5, we can see no visible peak. In fact, the only visible peak comes later between 17000 and 19000 messages in. Another example can be seen from Figure A.4 and Figure 4.4 where the peaks in Topic 4 analysed previously are not reflected in the Gini coefficient plot. We can put this down to how the Gini coefficient was calculated. The function was passed windows of the maximum topics in each message. This didn't take into account the value of the topic percentage, just whichever topic was the most prevalent at a given point. Therefore the Gini coefficient will spike when one topic is consistently higher than the others, even if only by a small margin. It can also miss smaller spikes which are only visible by plotting the topic percentage. Therefore it is important to use both techniques to fully understand the topic distributions in each stream.

5 Conclusion

Looking at the results from both quantitative and qualitative analysis we can see that chat engagement varies between the audiences of different streamers based on topic distribution and topic nature. It seems evident that the genre of game being played by the specific streamer is a clear reason for this.

Non-Shooters have a slightly less even topic distribution
The distribution of topics varies throughout the duration of a livestream. At some points the chat content is more broad, with several topics being mentioned. However, almost every stream has peaks where the Gini coefficient reaches margins up to around 0.6-0.7. In these moments, almost only one topic is present in the chat. We can make the conclusion that the topic distribution in non-shooter games is slightly less even than that of shooters, based on the median Gini coefficients of the 4 games analysed. This implies that the topic of conversation in the chats of non-shooters are more focused, with one topic the sole focus more often than in shooters were perhaps the chat content is more varied.

However, we cannot just take the Gini coefficient as our only factor. As seen by the plotting of Topic Percentages, there are peaks in topic percentage that the Gini coefficient fails to account for, and spikes in Gini coefficient that appear when there is seemingly no peak in topic percentage. Therefore it is important that both techniques were combined so that insights could be retrieved from both.

The genre of game being played affects the nature of the topics that cause peaks in chat engagement
What is similar between both shooters and non shooters is what the peaks in Gini coefficients represent. The nature of the dominating topic in each peak demonstrates that these peaks clearly indicate a stream highlight (see Figures 4.4, 4.5). Within the same genre, these topics are quite similar, however the content of these topics varies when compared between genres. This indicates two things: that audiences engage differently with livestream highlights depending on the genre of game they are watching and that topic nature is game genre dependent. This is particularly apparent with the non-shooters, where there the topics that cause the peaks have several overlapping words between the two games. However very few of these

words overlap with the topics causing the peaks in the shooters.

A correlation found between the shooter streams, Fortnite and CSGO, is the common theme of labelling opponents as "sweaty" or "try hard", a theme which was harder to locate in the non-shooter genre. We therefore could conclude that the audiences of shooter streams are perhaps slightly more aggressive, although the emphasis must be placed on "slightly" because this language could just as easily be said in a joke-like manner.

Why does chat engagement vary between different streamers based on the genre of the game?
It is hard to put reasons as to why the audiences of different genres behave differently, without generalising. The less evenly distributed nature of the non-shooter chat could potentially be put down to the non-shooter audience being more focused on the game than the other topics. Perhaps the age groups of the shooter and non-shooter audiences differ, although it seems irresponsible to say that the more violent shooter games wouldn't be watched by a younger audience anyway.

Future Work: Limitations and ESports relation to topic distribution
It should be noted that there are some potential limitations to the findings of this report. Some assumptions have been made which should be noted. Most notably these include the assumption that spikes in Gini coefficient relate to stream highlights and the assumed meanings of some emotes and phrases. Furthermore only a limited selection of streams were tested, and it may be the case that the genres "shooter" and "non-shooter" are too broad (particularly "non-shooter"). Therefore, future work in this area could work at resolving these limitations. More genres could be introduced, splitting the "non-shooter" label into several sub-genres and more streams could be analysed. There could be an investigation into the relation between peaks in Gini coefficient and stream highlights by looking at both the chat and stream footage if this was made available.
As a note for future research, it could be the case that games which are less accustomed to ESports have a calmer chat. ESports, short for Electronic Sports, is the name given to competitive gaming competitions [37]. The World of Warcraft stream was the only stream with no visible peaks in Topic Percentage (see Figure 4.7), despite a high Gini coefficient. World of Warcraft is a game which certainly has ESports events, although it is generally considered less suitable for this structure [38]. However the other games, Fortnite, CSGO and League of Legends are all very rooted in the world of competitive gaming [39] and each have much more visible peaks in topic percentage throughout their stream duration (see Figures 4.5, 4.4 and 4.6). It could potentially be the case then that chat engagement varies by the level of ESports involvement in a game, and this is certainly a topic which should be researched further.

A Appendix

Table A.1: Twitch Emotes and their meanings

Emote	Meaning/Usage
kekw	laughter
pepelaugh	laughter
monkaw	tense situation
trihard	"trying too hard", tense situation
kappa	irony and sarcasm
residentsleeper	boredom, late night stream, tiredness
monkahmmm	streamer is in deep thought, trying to decide their next move
4head	used in reaction to cheesy jokes and used in response to someone giving advice which is deemed obvious
weirdchamp	dissappointment or disbelief towards the streamer

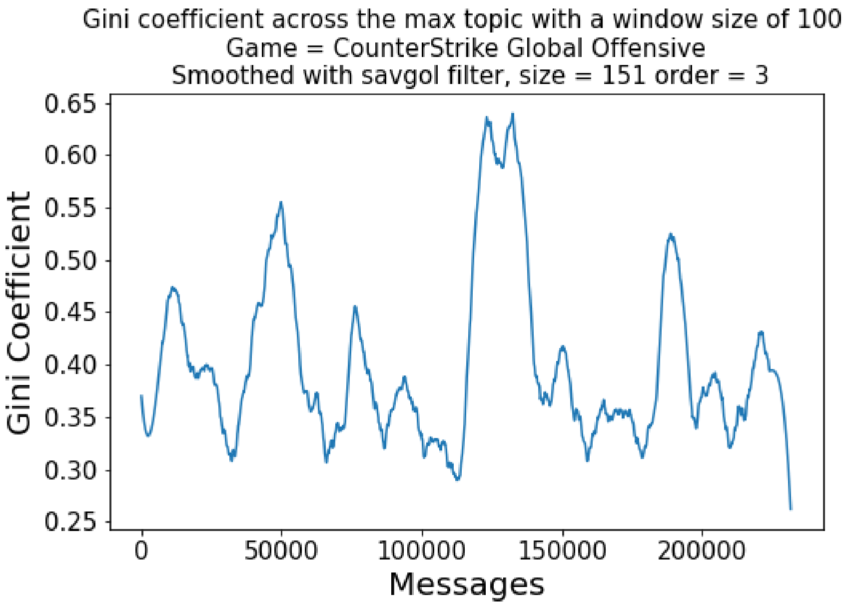

Figure A.1: CounterStrike Global Offensive Gini coefficient across stream duration

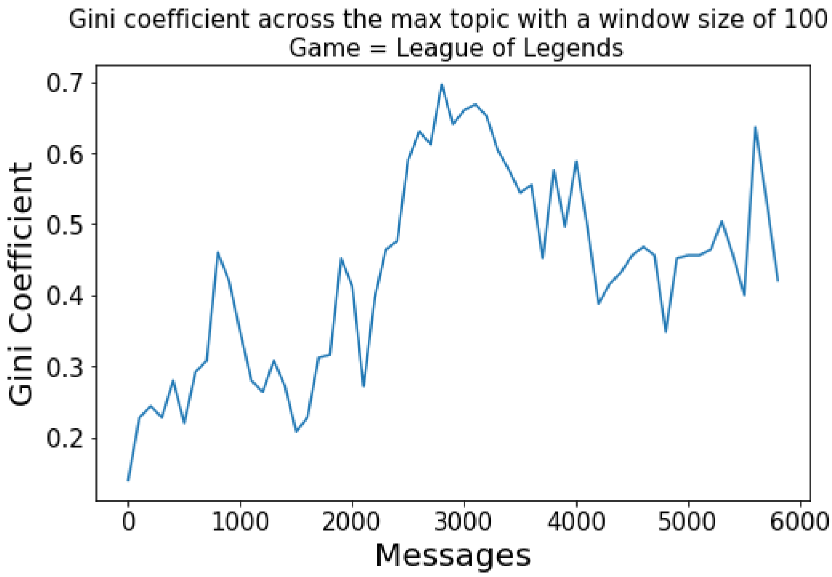

Figure A.2: League of Legends Gini coefficient across stream duration

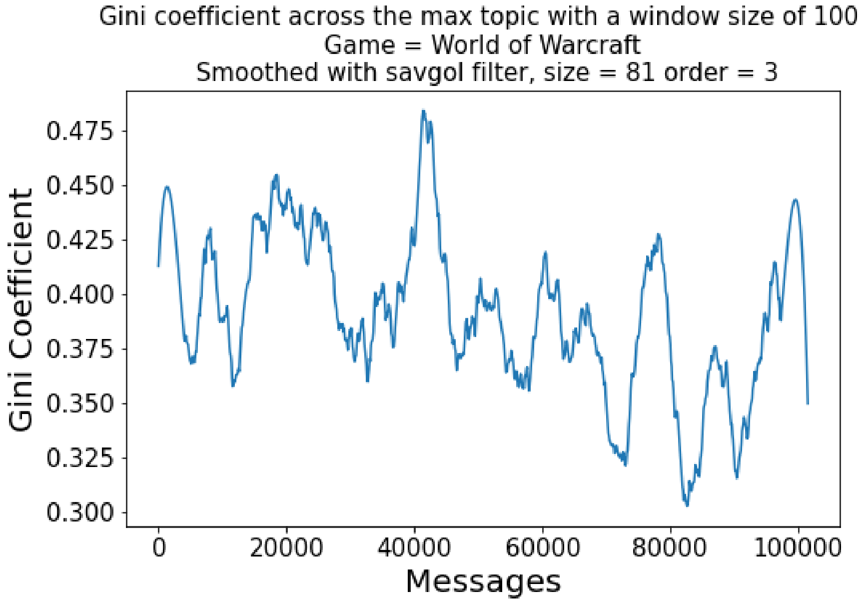

Figure A.3: World of Warcraft Gini coefficient across stream duration

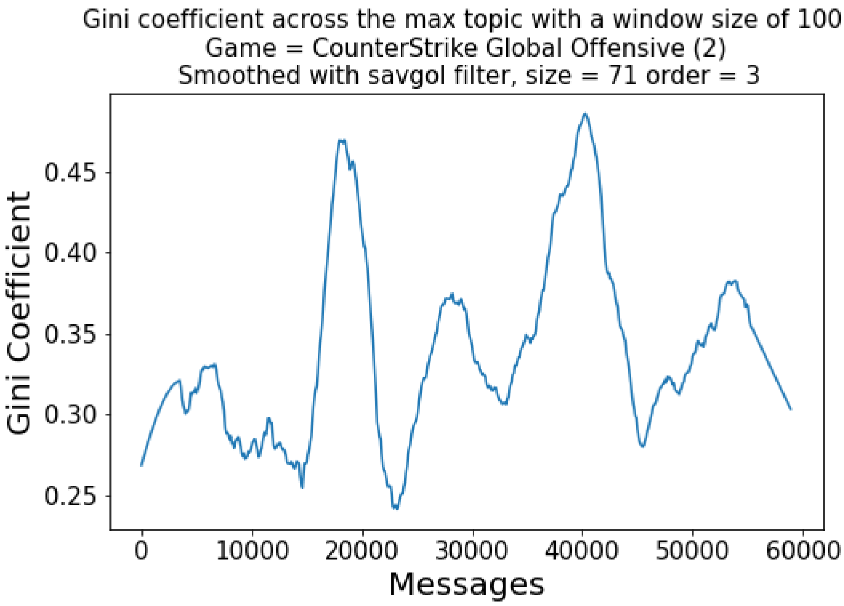

Figure A.4: CounterStrike Global Offensive Gini coefficient across stream duration. Second stream used to plot topic percentage

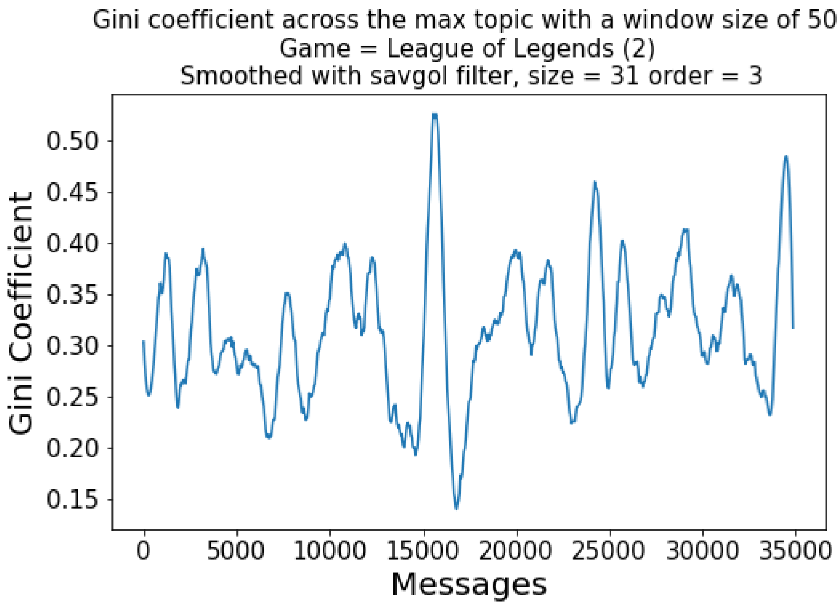

Figure A.5: League of Legends Gini coefficient across stream duration. Second stream used to plot topic percentage

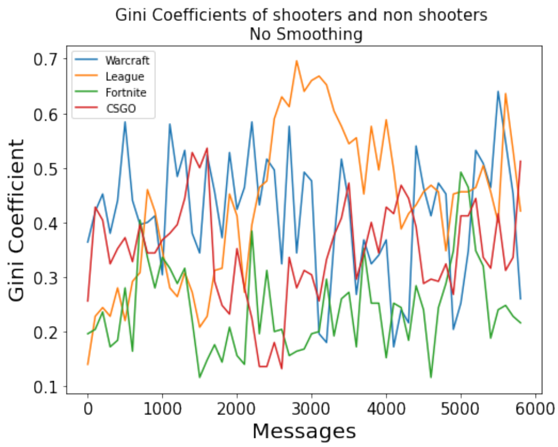

Figure A.6: Gini coefficient across stream duration without any smoothing applied

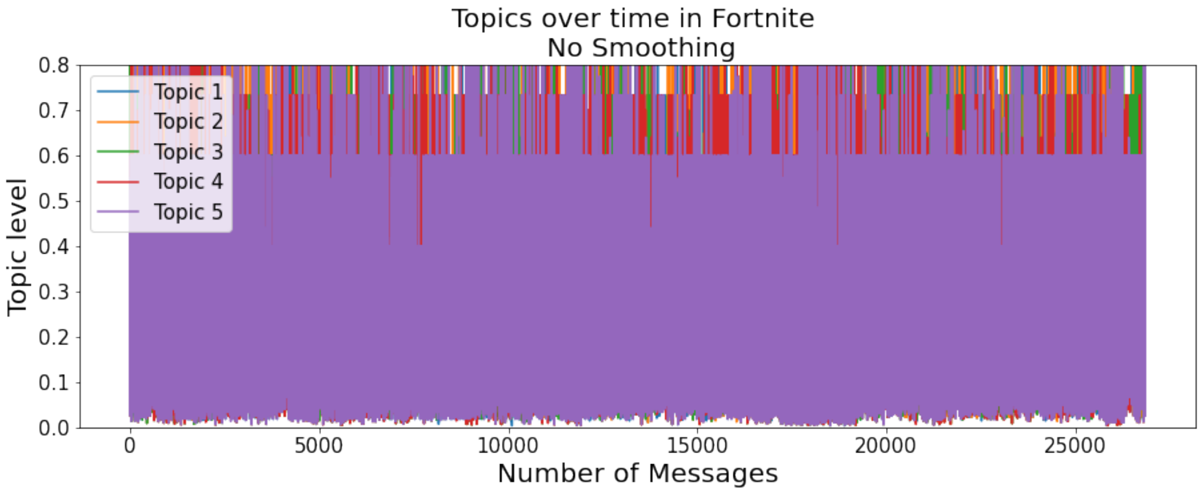

Figure A.7: Fortnite Topic Percentages over time without smoothing applied

Bibliography

[1] C. Ford, D. Gardner, L. Horgan *et al.*, 'Chat speed op pogchamp: Practices of coherence in massive twitch chat,' May 2017, pp. 858–871. DOI: 10.1145/3027063.3052765.

[2] C. Flores-Saviaga, J. Hammer, J. P. Flores, J. Seering, S. Reeves and S. Savage, *Audience and streamer participation at scale on twitch*, 2020. arXiv: 2012.00215 [cs.SI].

[3] M. Iqbal. 'Twitch revenue and usage statistics (2022).' (), [Online]. Available: https://www.businessofapps.com/data/twitch-statistics/ (visited on 02/05/2022).

[4] W. Lau. 'What is targeted advertising?' (), [Online]. Available: https://www.adroll.com/blog/what-is-targeted-advertising (visited on 02/05/2022).

[5] C. Ringer, M. A. Nicolaou and J. A. Walker. 'Twitchchat: A dataset for exploring livestream chat.' (), [Online]. Available: https://osf.io/39ev7/.

[6] S. Li. 'Topic modeling and latent dirichlet allocation (lda) in python.' (), [Online]. Available: https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24 (visited on 07/12/2021).

[7] R. Bhatia. 'Natural language processing for post-livestream analysis.' (), [Online]. Available: https://ronak-k-bhatia.medium.com/natural-language-processing-for-post-livestream-analysis-d3b5a8c77706 (visited on 07/12/2021).

[8] A. Hayes. 'What is the gini index?' (), [Online]. Available: https://www.investopedia.com/terms/g/gini-index.asp (visited on 29/04/2022).

[9] D. Bulygin, I. Musabirov, A. Suvorova, K. Konstantinova and P. Okopnyi, *Between an arena and a sports bar: Online chats of esports spectators*, 2018.

[10] D. Delfino. 'What is twitch?' (), [Online]. Available: https://www.businessinsider.com/what-is-twitch?r=US&IR=T (visited on 15/03/2022).

[11] twitch.tv. 'Guide to building a moderation team.' (), [Online]. Available: https://help.twitch.tv/s/article/guide-to-building-a-moderation-team?language=en_US (visited on 15/03/2022).

[12] H.-K. Han, Y.-C. Huang and C. Chen, 'A deep learning model for extracting live streaming video highlights using audience messages,' Dec. 2019, pp. 75–81. DOI: 10.1145/3375959.3375965.

[13] M. E. Roberts, B. M. Stewart and D. Tingley, 'Stm: An r package for structural topic models,' *Journal of Statistical Software*, vol. 91, no. 2, pp. 1–40, 2019. DOI: 10.18637/jss.v091.i02. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v091i02.

[14] T. Lebryk. 'Introduction to the structural topic model (stm).' (), [Online]. Available: https://towardsdatascience.com/introduction-to-the-structural-topic-model-stm-34ec4bd5383 (visited on 30/04/2022).

[15] S. Singh. 'Stemming in natural language processing.' (2020), [Online]. Available: https://www.c-sharpcorner.com/blogs/stemming-in-natural-language-processing (visited on 28/01/2022).

[16] wikipedia. 'Fortnite.' (), [Online]. Available: https://en.wikipedia.org/wiki/Fortnite (visited on 09/04/2022).

[17] ——, 'Counterstrike: Global offensive.' (), [Online]. Available: https://en.wikipedia.org/wiki/Counter-Strike:_Global_Offensive (visited on 09/04/2022).

[18] ——, 'League of legends.' (), [Online]. Available: https://en.wikipedia.org/wiki/League_of_Legends (visited on 09/04/2022).

[19] ——, 'World of warcraft.' (), [Online]. Available: https://en.wikipedia.org/wiki/World_of_Warcraft (visited on 09/04/2022).

[20] W. McKinney *et al.*, 'Data structures for statistical computing in python,' in *Proceedings of the 9th Python in Science Conference*, Austin, TX, vol. 445, 2010, pp. 51–56.

[21] texthero. 'Text preprocessing, representation and visualization from zero to hero.' (), [Online]. Available: https://texthero.org/ (visited on 15/03/2022).

[22] R. Rehurek and P. Sojka, 'Gensim–python framework for vector space modelling,' *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[23] S. Bird, E. Klein and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort *et al.*, 'Scikit-learn: Machine learning in Python,' *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,' *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[26] Esports.net. ‘Kekw meaning - the origins of the kekw emote.’ (), [Online]. Available: https://www.esports.net/wiki/guides/kekw-meaning-and-origins-of-the-kekw-emote/ (visited on 10/04/2022).

[27] own3d. ‘Pepelaugh emote meaning, origin more!’ (), [Online]. Available: https://www.own3d.tv/en/blog/dictionary/pepelaugh/ (visited on 10/04/2022).

[28] ——, ‘Monkaw emote meaning, origin more!’ (), [Online]. Available: https://www.own3d.tv/en/blog/dictionary/monkaw/ (visited on 10/04/2022).

[29] streamerfacts. ‘Trihard twitch emote meaning origin.’ (), [Online]. Available: https://streamerfacts.com/trihard-emote/ (visited on 10/04/2022).

[30] dictionary.com. ‘Kappa.’ (), [Online]. Available: https://www.dictionary.com/e/pop-culture/kappa/ (visited on 10/04/2022).

[31] Esports.net. ‘Residentsleeper meaning – the iconic emote of boredom.’ (), [Online]. Available: https://www.esports.net/wiki/guides/residentsleeper-meaning/ (visited on 10/04/2022).

[32] twitchemotes. ‘Twitch emotes - bringing a little kappa to you everyday.’ (), [Online]. Available: https://twitchemotes.com/channels/19571641 (visited on 10/04/2022).

[33] knowyourmeme. ‘Sweaty gamer.’ (), [Online]. Available: https://knowyourmeme.com/memes/sweaty-gamer (visited on 10/04/2022).

[34] StreamMentor. ‘Monkahmm meaning.’ (), [Online]. Available: https://streammentor.com/monkahmm-meaning/.

[35] getonstream. ‘What does the 4head emote mean on twitch? origin how to use!’ (), [Online]. Available: https://getonstream.com/what-does-the-4head-emote-mean/ (visited on 10/04/2022).

[36] T. Smith. ‘Weirdchamp meaning origin explained - twitch emote guide.’ (), [Online]. Available: https://www.streamergrowthschool.com/blog/weirdchamp-meaning (visited on 10/04/2022).

[37] M. Leroux-Parra. ‘Esports part 1: What are esports?’ (), [Online]. Available: https://hir.harvard.edu/esports-part-1-what-are-esports/ (visited on 11/04/2022).

[38] M. V. Stella. ‘World of warcraft esports in 2021: Does it have a future?’ (), [Online]. Available: https://esportsinsider.com/2021/08/world-of-warcraft-esports-in-2021-does-it-have-a-future/ (visited on 11/04/2022).

[39] EsportsEarnings. ‘Top games awarding prize money - esports game rankings :: Esports earnings.’ (), [Online]. Available: https://www.esportsearnings.com/games (visited on 11/04/2022).