Emote-Controlled

Obtaining Implicit Viewer Feedback through Emote based Sentiment Analysis on Comments of Popular Twitch.tv Channels

KONSTANTIN KOBS, ALBIN ZEHE, ARMIN BERNSTETTER, JULIAN CHIBANE, JAN PFISTER, JULIAN TRITSCHER, and ANDREAS HOTHO, Julius-Maximilians-University Würzburg, Germany

In recent years, streaming platforms for video games have seen increasingly large interest, as so-called "e-sports" have developed into a lucrative branch of business. Like for other sports, watching esports has become a new kind of entertainment medium, which is possible due to platforms that allow gamers to live stream their gameplay, the most popular platform being Twitch.tv. On these platforms, users can comment on streams in real-time and thereby express their opinion about the events in the stream. Due to the popularity of Twitch.tv, this can be a valuable source of feedback for streamers aiming to improve their reception in a gaming-oriented audience. In this work, we explore the possibility of deriving feedback for video streams on Twitch.tv by analyzing the sentiment of live text comments made by stream viewers in highly active channels. Automatic sentiment analysis on these comments is a challenging task, as one can compare the language used in Twitch.tv with that used by an audience in a stadium, shouting as loud as possible in sometimes non-organized ways. This language is very different from common English, mixing Internet slang and gaming-related language with abbreviations, intentional and unintentional grammatical and orthographic mistakes as well as emoji-like images called emotes. Classic lexicon based sentiment analysis techniques therefore fail when applied to Twitch comments.

In order to overcome the challenge posed by the non-standard language, we propose two unsupervised lexicon based approaches that make heavy use of the information encoded in emotes, as well as a weakly supervised neural network based classifier trained on the lexicon based outputs, that is supposed to help generalization to unknown words by use of domain-specific word embeddings. To enable better understanding of Twitch.tv comments, we analyze a large dataset of comments, uncovering specific properties of their language and provide a smaller set of comments labeled with sentiment information by crowd sourcing.

We present two case studies showing the effectiveness of our methods in generating sentiment trajectories for events live-streamed on Twitch.tv that correlate well with specific topics in the given stream. This allows for a new kind of implicit real-time feedback gathering for Twitch streamers and companies producing games or streaming content on Twitch.

We make our datasets as well as our code publicly available for further research.[1]

Additional Key Words and Phrases: Twitch, Sentiment Analysis, Feedback, Emotes

[1]Available at https://github.com/konstantinkobs/emote-controlled

Authors' address: Konstantin Kobs; Albin Zehe; Armin Bernstetter; Julian Chibane; Jan Pfister; Julian Tritscher; Andreas Hotho, surname@informatik.uni-wuerzburg.de, Julius-Maximilians-University Würzburg, Am Hubland, 97074, Würzburg, Germany.

2469-7818/2019/1-ART1 $15.00
https://doi.org/10.1145/3365523

ACM Reference Format:
Konstantin Kobs, Albin Zehe, Armin Bernstetter, Julian Chibane, Jan Pfister, Julian Tritscher, and Andreas Hotho. 2019. Emote-Controlled: Obtaining Implicit Viewer Feedback through Emote based Sentiment Analysis on Comments of Popular Twitch.tv Channels. *ACM Trans. Soc. Comput.* 1, 1, Article 1 (January 2019), 37 pages. https://doi.org/10.1145/3365523

1 INTRODUCTION

The gaming industry has become more popular in recent years and has developed into a highly lucrative economy branch [Nascimento et al. 2014]. Besides actively playing games, watching other people play games evolved into a new type of entertainment medium. Gamers are live streaming their gaming sessions on certain platforms, while other people can watch them and comment on the events in the stream in real-time. The most popular of these game streaming platforms is Twitch.tv[2], which has even turned into one of the largest Internet traffic generators in the US [Zhang and Liu 2015]. As comments on a specific stream event follow closely on the event itself, sentiment based trends shown in the comment section of the stream can give valuable feedback to the streamer, who can correlate the trends with actions, statements, or other events happening in the stream. This enables streamers to adapt their behavior or presentation in real-time, or learn for future streams in order to achieve the desired emotions from the audience. Due to popular streams getting many comments per second, automatically estimating the comments' emotions in real-time would facilitate this implicit way of gathering feedback. In this paper, we automatically assess the emotion of comments by applying sentiment analysis methods in highly active streams. This way it is possible to check whether an event is positively or negatively perceived by commenting viewers, which helps streamers understand the preferences of their target audience.

The biggest challenge in performing sentiment analysis on Twitch comments is the non-standard language. An impression of Twitch language usage can be seen in Figure 1. The language consists of many abbreviations, intentional and unintentional grammatical and orthographic mistakes, duplicated phrases, and short sentences. Pictographical images and animations called emotes are also very popular[3] due to their ability to express emotions in a way that is easily interpretable by the human eye. The way emotes are used makes them an own form of language that is captured in the comment by the emote's, sometimes cryptic, text representation. For this reason, lexicon based sentiment analysis methods designed for common English typically fail to correctly classify Twitch comments.

In this work, we explore the suitability of emotes as emotion indicators to perform sentiment analysis on Twitch comments, and introduce multiple methods that rely on emote-, emoji-, and word-sentiment lexica. We show that emotes are a good complement to other lexica. Additionally we compare two types of lexica: *Average based* sentiment lexica that provide one sentiment score per word and *distribution based* sentiment lexica that contain a distribution over all classes based on the annotator's votes. We show that distribution based sentiment lexica improve our test scores as they provide more information regarding "controversial" emotion indicators, that is words that can have positive as well as negative connotations.

Our contributions in this paper are threefold:

(i) We show that emotes can be used as additional emotion indicators in a lexicon based sentiment analysis approach to classify Twitch comments more reliably.
(ii) We show that the sentiment of Twitch comments correlates with events in the stream, allowing Twitch streamers to acquire implicit feedback from the gaming community.

[2]https://www.twitch.tv
[3]https://stats.streamelements.com/c/global

(iii) We provide an unlabeled dataset of Twitch channel chat logs, a labeled sample of Twitch comments with their respective sentiment polarity, as well as introduce first analyses and sentiment classification methods to encourage further scientific research on this data.

The remainder of this paper is structured as follows: Section 2 gives background information on Twitch and emotes. Section 3 introduces the data we used in our experiments, the procedure to obtain a labeled sentiment analysis dataset from this data, and the sentiment lexica we utilized in the development of our approaches. Section 4 then analyzes the obtained unlabeled and labeled datasets. In Section 5 we describe our methods and results, as well as provide baseline approaches. Section 6 gives thorough insights into the results and the differences between our methods. We then take our methods to the test by providing two case studies. In these studies described in Section 7 we qualitatively and quantitatively measure the ability of our methods to gather implicit feedback on real world streams. A critical discussion of our methods and the results is given in Section 8. Related work is provided in Section 9, followed by a conclusion of the work in Section 10.

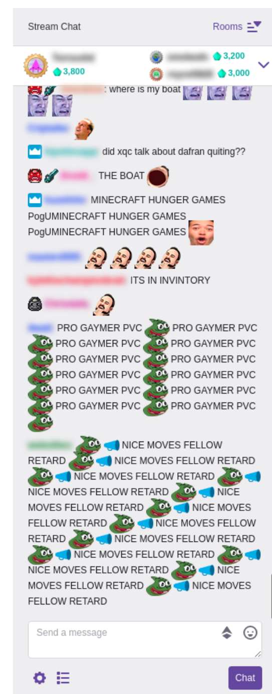

Fig. 1. Screenshot of a Twitch comment section. The language used in the comments is fairly different from common English. In the bottom right, an emote picker helps with selecting an emote. User names are anonymized due to privacy reasons.

2 TWITCH.TV

In this section we give an overview on Twitch and its historical as well as cultural background. We will introduce information about Twitch in general, in order to understand the platform and circumstances in which Twitch comments are written. As we will show in more detail in Section 3.1, the language of Twitch comments is very different from common English. A crucial part of the language on Twitch are emotes, which we will also introduce in this section.

2.1 Overview

Twitch.tv is a live streaming platform that allows companies and individuals to broadcast live entertainment content. Every user has a profile page which is called a "channel", on which they can live stream any time. In addition to that, channel owners can save their streams to the "Videos" section of the channel, where users can watch them on demand. Users can follow channels, resulting in an easily accessible sidebar entry. A list always shows the currently streaming channels the user follows. To support the channel owner monetarily, users can also subscribe to a channel for a monthly fee, which grants subscribers access to channel specific emotes that can be used everywhere on Twitch.

Regardless of whether the streamer is currently streaming, users can live chat using the channel's comment section. Every channel has a main chat room called "Stream Chat" and can possess additional named chat rooms. Channel owners can nominate other users to be so called "moderators" of the channel, meaning they are allowed to curate a chat by deleting comments or ban users from commenting on the channel altogether. Chat rooms can be set to be accessible only to subscribers or moderators, for example to discuss moderation related matters or to create motivation for users to subscribe to the channel.

2.2 History and Culture

Launched in 2011 as a spin off to the multipurpose streaming platform Justin.tv, Twitch was initially branded as a platform for broadcasting competitive esport.[4] Twitch itself soon overtook its parent Justin.tv in popularity, which resulted in the company shutting down Justin.tv and focusing solely on Twitch in 2014.[5] Soon after, Twitch was acquired by Amazon.[6] As of July 5th, 2019, Twitch is rank 12 in the USA on the Alexa Rank and rank 25 globally regarding visitor counts.[7]

Live content on Twitch ranges from simple "Let's plays", that is, streamers broadcasting themselves playing a game and commenting their gameplay, to the live streaming of large events such as esport competitions or video game press conferences. Recently, "real life" content has also been increasing, which includes streamers broadcasting themselves cooking, exploring cities and nature, or just talking and interacting with their viewers. For many individual streamers, Twitch has also become a source of income via donations and subscriptions. Given the information from Twitchstats[8], as of June 2019, the most subscribed Twitch streamer "shroud" earns approximately 175,000 $ per month from subscriptions. Additionally, advertising deals and branded content are another big source of income for streamers and Twitch. By advertising through online personalities, companies can reach their target audience more directly than anywhere else. In the case of Twitch, this is mostly important for video game companies as viewers on Twitch have high affinity for gaming, Internet, and related topics. Due to monetary and research interests, streamers or Twitch may want to analyze the sentiment of users regarding certain products or presentations. The resulting information can then be used to better suit a broader audience and therefore increase user engagement and income.

2.3 Emotes

Twitch emotes are named little pictures or animations available to Twitch users in the comment section next to streams. They are an essential part of the language on Twitch, as they enable users to quickly express specific reactions without writing verbose texts. In this work, we use the term "emote" exclusively for Twitch emotes in the form as they are explained in this section. We distinguish between emotes and unicode emojis that are used for example in messaging apps and are available on multiple platforms.[9]

Every emote has its own meaning, back story, and use cases. While some emotes' meanings can be inferred by looking at the image representation, others may not be easily understood by people unfamiliar with Twitch. One of the best known examples is the emote Kappa, which evolved to denote sarcasm when used at the end of a sentence.[10] An example would be the comment "Well played! ", where the commenter actually thinks that the streamer has made some grave mistake. Table 1 shows examples of emotes and their usage.

Twitch emotes often depict popular streamers (for example PogChamp showing professional *Street Fighter* player Gootecks or 4Head showing *League of Legends* streamer Cadburry), (former) Twitch.tv/Justin.tv employees (for example Kappa depicting Josh DeSeno), fictional characters (for example FeelsGoodMan utilizing Matt Furie's *Pepe the Frog*), or refer to popular videos

[4] https://www.businesswire.com/news/home/20110606005437/en/Justin.tv-Launches-TwitchTV-World%E2%80%99s-Largest-Competitive-Video
[5] https://www.theverge.com/2014/8/5/5971939/justin-tv-the-live-video-pioneer-that-birthed-twitch-officially-shuts
[6] https://blog.twitch.tv/a-letter-from-the-ceo-august-25-2014-b34c1cfbb099
[7] https://www.alexa.com/topsites/countries/US (Retrieved: July 5, 2019)
[8] https://twitchstats.net/real-sub-count/2019/June
[9] "UCD: Emoji Data for UTR #51". Unicode Consortium https://unicode.org/Public/emoji/11.0/emoji-data.txt
[10] https://knowyourmeme.com/memes/kappa

Table 1. Examples of Twitch Emotes

Emote	Name	Meaning
	Kappa	Denotes sarcasm of the previous text if used at the end of a sentence.[a]
	PogChamp	Amazement for example if the streamer shows extraordinary skill in the game[b]
	LUL	General laughter or amusement (see relation to the abbreviation “lol”)[c]
	WutFace	Disgust[d]

[a] https://knowyourmeme.com/memes/kappa
[b] https://knowyourmeme.com/memes/twitch-emotes
[c] https://knowyourmeme.com/memes/lul
[d] https://knowyourmeme.com/memes/twitch-emotes

(for example  haHAA showing Andy Samberg’s face from a sketch music video) and are used, like emojis, to express certain feelings or emotions in the context of the currently airing stream.

Emote names are substituted with the corresponding image representation in the chat if the emote is available for the logged in user. Emote names are case-sensitive and need to be typed correctly to be converted to the corresponding emote image. To prevent mistakes and to facilitate the selection of emotes, a list of available emotes similar to emoji pickers on smartphones is also available in the comment section.

Twitch itself currently offers around 250 global emotes[11] which can be used by every logged in user. Additionally, channels can offer a varying number of subscriber emotes depending on their popularity. These are only available for viewers with paid subscriptions to the channel but can be used by those in chats of other channels, if acquired. The image of a non-available emote can still be seen by others if used by a user for which it is available. In total, there are more than 1,100,000 subscriber emotes on Twitch.[12]

Another way to display emotes in chat are (browser) extensions such as “Better Twitch TV” (BTTV)[13] or “FrankerFaceZ” (FFZ)[14]. Among other functionality, these extensions introduce new emotes which are available to everyone using the extension. In contrast to Twitch’s own emotes, these can only be seen if the extension is installed. In a survey among Twitch users (see Section 3.3.3), 60 % of participants were using BTTV or similar extensions.

BTTV offers around 100 new emotes[15] and FFZ offers approximately 165,000 public emotes and 60,000 private emotes[16]. FFZ emotes can be created by users and added to the database. Twitch streamers can then choose to add emotes to their channel which results in a substitution of the emote name with its corresponding image. While any streamer can add public FFZ emotes to their channel, private FFZ emotes can only be used if the uploader of the emote agrees.[17]

[11] https://twitchemotes.com
[12] Estimated using https://twitchemotes.com/apidocs (Retrieved: July 5, 2019)
[13] https://www.nightdev.com/betterttv/
[14] https://www.frankerfacez.com/
[15] https://nightdev.com/betterttv/faces.php
[16] https://www.frankerfacez.com/emoticons/?q=&private=on&sort=created-desc (Retrieved: July 5, 2019)
[17] https://www.frankerfacez.com/terms

3 RESOURCES

In this section, we describe the resources used in this paper. The first of these resources is a large unlabeled dataset of Twitch comments that we crawled from Twitch. Next, we introduce the procedure we used to manually label parts of this dataset with sentiment information as well as the resulting labeled dataset. Finally, we describe the three sentiment lexica we use in this work, two existing ones and a novel emote sentiment lexicon that we created by crowd sourcing.

3.1 Unlabeled Twitch Comments Data

For the analysis of the Twitch domain, we have collected a large dataset of publicly accessible "Stream Chat" comments from Twitch.tv. For this, we periodically queried the official Twitch API for current live streams. Distributed crawlers then join or leave channels depending on their streaming status and subscribe to new comments. These comments are then deduplicated, enriched with metadata, and saved. For this work, we focus our analysis on three months, namely April, May, and June of 2018.

We collected 998,102,078 comments for April, 1,093,323,667 comments for May, and 977,608,889 comments for June, leading to a total dataset size of 3,069,034,634 comments.

Table 2 shows the information that is contained in this dataset for an exemplary comment.

Table 2. Information attached to one comment in the dataset

Column	Example	Explanation
date	2018-05-05T04:49:53.602Z	The UTC timestamp of the comment
channel	moonmoon_ow	The channel the comment was made in
game	Darkest Dungeon	The game that was streamed during the publication of the comment
user	user1234	The commenting user's user name. User names are anonymized due to privacy reasons in this example and in the publicly available dataset.
mod	False	Whether the commenting user is a moderator of the current channel
subscriber	True	Whether the commenting user is a subscriber to the current channel
message	you can do it moon moon2CUTE Clap2 moon2S Clap2 moon2A Clap2 moon2N	The comment's text including all emote text representations

3.2 Labeled Twitch Comment Data

In order to evaluate our sentiment analysis methods, we need a dataset that has been manually annotated with sentiment information. To this end, we created a dataset to be labeled in a crowd sourcing campaign on figure-eight[18]. We selected the five most commented English Twitch channels from May 2018 that are not dominated by bots: We used the channels forsen, moonmoon_ow,

[18] https://www.figure-eight.com

riotgames, sodapoppin and xqcow. These highly active channels are especially interesting for automatic analysis, as they receive comments in a frequency that makes it impossible for the streamer to read new comments in real-time. From this dataset consisting of around 14.4 million comments we sampled 2000 comments. We used a weighted sampling scheme instead of sampling uniformly for the following reasons: The majority of comments on Twitch.tv consist of only very few words (see Section 4.1), making them targets of low interest for human annotations. Comments that only consist of one word are also most likely to be present in the sentiment lexica we present in the next subsection. This makes estimating the sentiment of such comment trivial. Additionally, comments consisting of only few unique tokens that are repeated many times do not contain enough information to manually estimate the sentiment of the comment. Long comments that only contain one word multiple times are also captured by a simple lexicon lookup. When later using our methods to analyze sentiment trends in Twitch streams, labeling comments with one unique word is mostly trivial. Therefore, with this sampling process, the goal is to find comments that consist of more than one word and contain enough words for a human to label. This allows for a sample that is not directly covered by the lexicons. We weighted every comment from the dataset using the following formula for sampling:

$$\text{weight} = \frac{\#\ \text{unique words}}{\log(\#\ \text{words} + 1)}$$

This weighting ensures that comments with a higher number of unique words are sampled more often, while at the same time not simply selecting the longest comments because of the normalization over the number of tokens in the comment. Even though this process may lead to non-representative samples of the complete Twitch comments corpus, human raters can better estimate the sentiment of such comments without context, which is crucial for a valid evaluation of sentiment analysis methods. In our case studies in Section 7, we show qualitatively that the methods that are evaluated on this sampled dataset are capable of capturing the sentiment trends of highly active Twitch streams.

The sampled 2000 comments were then given to crowd workers, where each comment was rated by three workers. To ensure that the workers provided annotations to their best knowledge instead of randomly selecting answers, we included some *control questions* in the dataset, where the comments had previously been labeled by domain experts. Crowd workers with too many incorrect responses to these control questions were excluded from the crowd sourcing job.

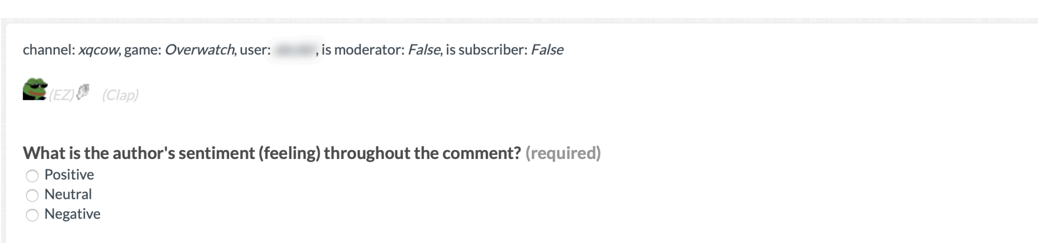

Fig. 2. An excerpt of how the crowd workers saw the job. Emote names were converted to their equivalent images if possible, while the names were put in parenthesis behind them. Additionally, meta-information about the comment was given, such as the channel, the game, the user (blurred for privacy purposes in this work), whether the user was a moderator of the current channel, and whether the user was subscriber to the current channel.

An excerpt of the questionnaire's interface is displayed in Figure 2. The workers were given the comment as well as some meta-information about the comment. They were then asked the following question: "What is the author's sentiment (feeling) throughout the comment?" The possible answers were "Positive", "Neutral", and "Negative".

In order to convey the emotes' meaning, replicating the appearance of emotes on the crowd sourcing page as closely as possible to Twitch's appearance was important. To this end, we added the image representation of the emotes to the view shown to crowd workers. Next to each emote image, the emote name was also displayed in parentheses, as the identifier may have an impact on the understanding of the emote meaning. Some of the emote images were not available via the used APIs, so we requested the crowd workers to use Internet search engines when encountering unknown words or identifiers.

The resulting dataset has an inter-annotator agreement of 0.497, measured by Fleiss' Kappa [Fleiss 1971], given the three categories and the three annotations per comment. Comparable work such as [Narr et al. 2012] and [Basile and Nissim 2013], which were based on tweets labeled with three classes and by three annotators, reported an inter-annotator agreement of 0.407 and 0.397, respectively. Therefore, even though Twitch comments are usually very short and do not follow the common rules of English grammer, the agreement between crowd workers was moderate.

53.35 % of comments were labeled with the same sentiment polarity by all three annotators. For 96.1 % of the comments, at least two annotators agreed on the same category, allowing majority voting for the comment's sentiment. We evaluated our approaches only on these 96.1 % of comments, that is on 1922 comments. Fleiss' Kappa increases to 0.5379 when considering only the selected comments for evaluation.

From these comments, 404 (21.02 %) were classified as negative, 748 (38.92 %) as neutral, and 770 (40.06 %) as positive.

3.3 Sentiment Lexica

Sentiment lexica are a commonly used resource in sentiment analysis. Generally, they are lists of words associating each word with a polarity, providing valuable hints for the sentiment conveyed by sentences including these words. We used three lexica for assessing the sentiment of Twitch comments, namely a word-, emoji-, and emote-based lexicon. While there is a rich amount of word-based and emoji-based lexica, to the best of our knowledge, there exists no sentiment lexicon for emotes, which is why we created one using crowd sourcing. In this section, we describe the lexica we used in our work, as well as the procedure we used to create the emote sentiment lexicon.

For the construction of sentiment lexica, it is common to collect labels from multiple annotators for each word and aggregate these ratings. This can be done in two ways: averaging the individual scores (*averaging approach*) or building a distribution over the labels (*distributional approach*), resulting in lexica that, in the following, we call L_{avg} and L_{dist}, respectively. We argue that the latter approach is more reasonable, as it provides the ability to accurately represent "controversial" words: Take for example a word that is labeled as negative by 5 annotators, as neutral by 0 annotators and as positive by 5 annotators. Averaging the scores would assign the word an overall score of 0, that is, neutral. On the other hand, representing the word by the distribution $(5, 0, 5)$ preserves the information that the word can be either positive or negative, but never neutral. Since we want to incorporate this information into our classifiers, we selected sentiment lexica where the distribution over labels is available, further described in the following subsections.

3.3.1 VADER Lexicon. The VADER lexicon is a word based sentiment lexicon [Gilbert 2014]. It provides a list of 7517 English words, phrases and ASCII text emoticons (for example ":)" or ":P").

Every entry was rated by ten subjects on an integer scale from -4 (very negative) to 4 (very positive).

The individual labels are available as part of the dataset, enabling us to use both the averaging and the distributional approach described above. For the former, we normalized the values to the range $[-1, 1]$ before averaging. For the latter, we grouped the scores from -4 to -2 as negative, -1 to 1 as neutral and 2 to 4 as positive and constructed the distribution over these labels.

3.3.2 Emoji Lexicon. To account for unicode emojis, the emoji sentiment lexicon from [Kralj Novak et al. 2015] was used. It contains 969 unicode emojis and their respective sentiment distribution based on the sentiment of tweets these emojis appear in. Again, we can construct both average and distributional labels from this lexicon.

To ensure reliable labels, we only considered the emojis that appear in 50 or more tweets, which yields annotations for 300 unicode emojis.

3.3.3 Emote Lexicon. Since emotes are of special importance in Twitch comments, we created our own sentiment lexicon for emotes. As labeling all emotes is not feasible, we selected the top 100 emotes measured by the usage frequency in the unlabeled dataset.

To label these emotes, a survey was conducted using Google forms. The survey was published on two gaming-related Twitter accounts and on various gaming and Twitch related subreddits on Reddit[19] to ensure that mainly users of Twitch and therefore people with background knowledge about emotes and emote usages were participating. Questions about the familiarity with Twitch and Twitch emotes as well as the preferred use of Twitch (browser) extensions were asked at the beginning.

Participants were then shown images and text representations of the emotes, including Twitch emotes as well as BTTV and FFZ emotes. The task was to "rate [the emotes] as either negative, neutral or positive, according to the sentiment of the situation in which you would or already have used this emote.". The answer to unknown emotes should be left blank.

In total, the survey received answers from 108 participants, which was sufficient to show clear tendencies for the sentiment of most emotes. Table 3 shows emotes with the most and least answers as well as an example for an emote that has no clear tendency in sentiment to show that not all emotes can clearly be put into one category. The full survey results can be found in the supplemental material.

A majority of participants were already acquainted to Twitch and Twitch emotes. Approximately 80 % of participants stated to be "fairly familiar" or "extremely familiar" with Twitch emotes and how they are used ("not at all familiar": 1.9 %, "slightly familiar": 4.7 %, "moderately familiar": 13.1 %, "fairly familiar": 37.4 %, "extremely familiar": 43 %). Approximately 60 % of participants stated that they use Twitch enhancing (browser) extensions such as BTTV.

Again, we used both the averaging and the distributional approach to create sentiment lexica from this survey.

4 DATASET ANALYSIS

In order to build a successful sentiment analysis model for Twitch comments, it is necessary to understand the peculiarities of their language. To this end, we provide a thorough analysis of our data in this section. We find that the language used in Twitch comments differs strongly from standard English and from messages on Twitter in multiple ways. In particular, we show that emotes are a crucial component of the comments and must be given special attention.

We start by an analysis of the unlabeled dataset and then move on to the labeled dataset.

[19] http://reddit.com

Table 3. Answers to Twitch Emote Sentiment Survey

Emote \ Sentiment	Negative	Neutral	Positive	Unknown/NA
FeelsBadMan	71	17	19	1
FeelsGoodMan	1	7	98	2
LUL	11	23	72	2
OMEGALUL	17	26	62	3
PogChamp	1	3	101	3
⋮	⋮	⋮	⋮	⋮
Jebaited	25	27	37	19
⋮	⋮	⋮	⋮	⋮
mcaT	10	34	12	52
forsenPls	13	26	17	52
PepoDance	9	26	20	53
RedCoat	5	46	4	53
jinnytHype	7	31	17	53

4.1 Unlabeled Data Analysis

We analyzed several characteristics of the unlabeled dataset introduced in Section 3.1, including the mean length of comments and the most frequently used words and emotes.

Comment Length. A mean comment length of approximately 5.12 words, with a minimum of 1 word, maximum of 250 words, median of 3 words and standard deviation of approximately 6.07, indicates that a typical Twitch comment is fairly short. Approximately 29 % of all comments consist of only one word. This can be explained by the fast pace at which users create comments. Short comments can be typed and submitted faster, thus having an advantage for reacting to an event happening in the stream and having more time following the actual stream.

Another common practice of Twitch users is to create comments that are constructed by duplicating a comment text multiple times (see Figure 1 for examples). This behavior leads to comments that are rather long, and therefore visible in the fast-moving chat, while still being very fast to type. To show this empirically, we analyzed the number of unique words per comment in relation to the length of the comment, revealing a mean of 4.61 unique words with a minimum of 1 unique word, maximum of 243 unique words, median of 3 unique words and standard deviation of approximately 5.04. Plotting the number of words for a comment versus the number of unique words for that comment illustrates the aforementioned behavior, which is shown in Figure 3.

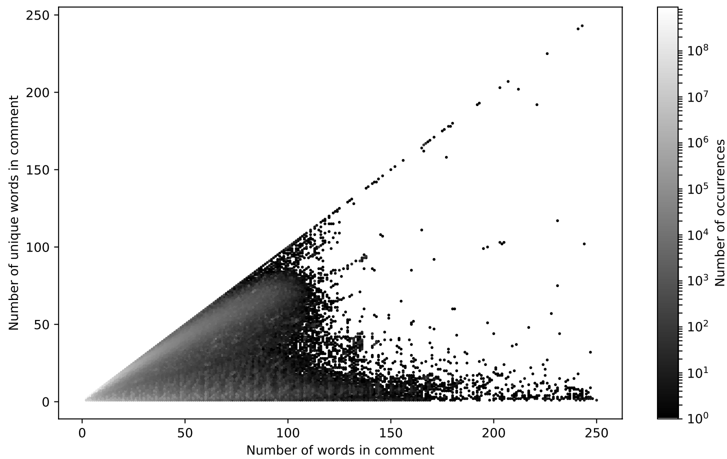

Fig. 3. The relation between number of words and number of unique words per comment. The more comments in the dataset have the same combination, the lighter the color of the dot. Two trends (lighter areas) regarding this relation can be identified.

Most of the comments consist of only a few words. Apart from that, there are two trends visible in the figure: The lighter upper trend follows Heaps' law [Heaps 1978], which is a typical words-to-unique-words ratio function that can be found in natural language documents and texts. However, the lower trend shows that there are many comments that are relatively long while consisting of only a few unique words, thus exhibiting the behavior described above. This means that on Twitch, commenters are trying to get attention by creating comments that are as long as possible with little effort. This behavior can be compared to fans in a stadium that are trying to drown fans of the opposite team by making noise. We are not aware of other corpora that contain natural language with this kind of linguistic specialty, as other corpora have shown mostly perfect fit to Heaps' law [Loreto et al. 2016].

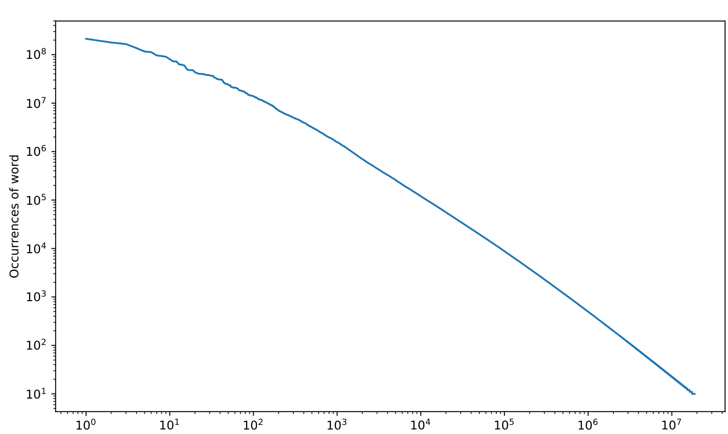

Fig. 4. Rank-frequency plot of the Twitch comments' vocabulary

Most Frequent Words. The rank-frequency plot for words in the unlabeled dataset is depicted in Figure 4. The data follows a Yule-Simon distribution [Simon 1955], which is the result of a preferential attachment process, also called Yule process [Yule et al. 1925]. In the case of Twitch's comment vocabulary, this stochastic process works as follows: There is a growing number of words in the vocabulary. Yule's process then states that the further usage of the words depends linearly on the number of usages before, that is words that are already used very often are going to be used very often in the future. This is similar to observations in word frequencies that appear in speech transcriptions. When people speak, they mostly use words that they are familiar with, which leads to word frequencies that are biased towards most frequent words [Lin et al. 2014]. Twitch seems to be a platform in which people tend to comment in a way that resembles speech-like patterns, which fits the informality of commenting Twitch users' behavior, as shown in the previous paragraph.

Our claim that the language used on Twitch is very different compared to common English is further supported by analyzing the 20 most frequent words in the unlabeled dataset. For this analysis we removed stop words from multiple languages (English, Portuguese, Spanish, German, Russian), as the Twitch community is international and some big channels are mainly commented by non English speaking users.

The resulting most frequent words are shown in Table 4. In comparison to that, the top 20 words from English[20] and Twitter[21] without stop words are shown in the other columns. Half of Twitch's top 20 words that are not stop words are emotes. Even when excluding emotes, the vocabulary used in Twitch comments seems to be quite different from common English, as both top 20 lists only share 4 words. The difference to Twitter is a lot smaller with both lists sharing 11 words. In contrast to both lists, Twitch also contains common Internet slang words such as "u" as a shorter term for "you" or the emoticon "xD" as an expression of laughter. Numbers seem to be used relatively often for multiple reasons: First of all, they can be typed faster than corresponding words (for example, "two" and "to" become "2"). Regarding game streams on Twitch, many games have countable items that are commented on (for example "just get 1 stick and 2 diamonds"). Numbers are also used as a simple interactive form of polling in the Twitch comment section, where users just comment with the number corresponding to the option that they agree with. Words like "game", "stream", or "play" indicate the gaming domain that Twitch is in. Together with words like "like" and "good", this indicates that Twitch comments are used to express user's sentiment related to events happening in the current stream more frequently than for general discussion.

Channel Activity. While there are approximately 700,000 different channels recorded in the overall dataset, not all receive comments on a regular basis. Looking at the three months separately, in each of the months 350,000 to 400,000 channels received at least one comment. Requiring the channel to receive at least one comment in every of the three months, this number reduces to approximately 150,000 channels. Figure 5 shows the number of comments across channels for the entire dataset. The plot is divided into three segments. The head (up to first dotted vertical line) follows the power law, as the curve is approximately linear in the log-log-plot. The middle (first to second dotted vertical line) follows the power law as well, however, with a different slope. For the tail, the number of comments per channel decreases drastically.

This phenomenon was already observed in other works analyzing YouTube and Netflix videos regarding their respective number of ratings and views, which is a similar scenario to Twitch channels and their respective number of comments. All of them conclude that users on such media sites discover content by search rather than by browsing, which makes less popular items harder to discover. This way, already popular users receive even more user views and comments as they

[20] https://github.com/first20hours/google-10000-english/blob/master/google-10000-english.txt
[21] According to http://techland.time.com/2009/06/08/the-500-most-frequently-used-words-on-twitter/

Table 4. Lists of most frequently used words. All of them exclude common stop words. Underlined words are shared between Twitch and common English, bold words between Twitch and Twitter.

Rank	Twitch	Twitch (no emotes)	English	Twitter
1	LUL	**like**	new	tinyurl.com
2	Kappa	**get**	home	new
3	<3	lol	us	**like**
4	PogChamp	u	page	**good**
5	**like**	**good**	search	**get**
6	**get**	**2**	free	**time**
7	lol	1	**one**	day
8	:D	game	information	**one**
9	Kreygasm	stream	**time**	twitter
10	Clap	**got**	site	going
11	u	**one**	may	**go**
12	**good**	**go**	news	rt
13	:)	play	use	**know**
14	**2**	xD	see	today
15	1	3	contact	love
16	game	**know**	business	work
17	HeyGuys	**time**	web	**got**
18	BibleThump	**think**	also	**2**
19	stream	see	help	**back**
20	**got**	**back**	get	**think**

are found via search, other users, or the Internet, while lesser-known users are not getting any new audience members [Cha et al. 2007; Cheng et al. 2008; Halvey and Keane 2007].

In fact, 5 % of all comment activity is accounted for by the top 29 channels in the recorded three months. This means that the most active channels are highly influential on the overall most used words and emotes. Meanwhile, the average number of comments per channel in the recorded three months period is approximately 4421 comments (standard deviation: approximately 54,891, minimum: 1 comment, maximum: 13,331,333 comments, median: 63 comments). The top 10 most commented channels are forsen, sodapoppin, xqcow, hanryang1125, yapyap30, moonmoon_ow, saddummy, twitchmedia_qs_10, yoda, and greekgodx. From these channels, hanryang1125, yapyap30, and saddummy are Korean and yoda is Portuguese, the remaining channels are English. twitchmedia_qs_10 is the official Twitch stress-test channel: on this channel, mostly bots

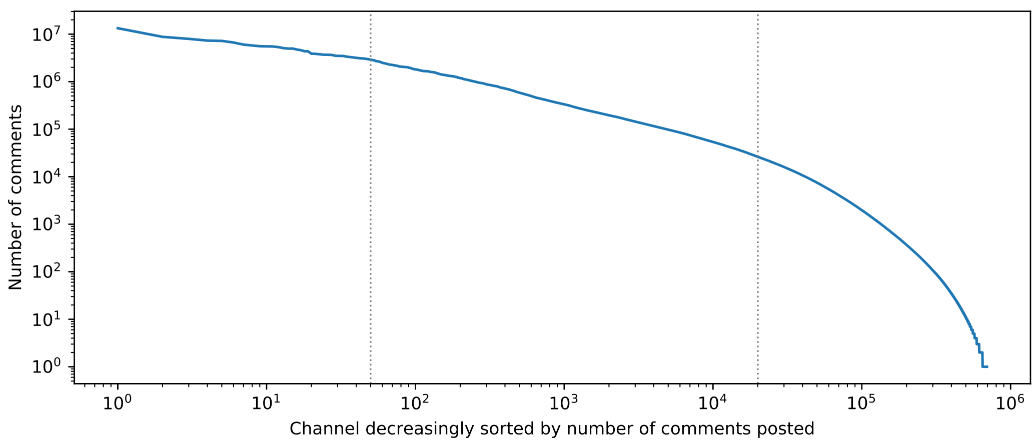

Fig. 5. Number of comments per channel.

produce the comments, which means that there is no human intention found in this channel's comments.

Usually, streamers do not stream continuously. Users can comment on a channel whenever they want, however, while streaming, the comment frequency is usually higher than when the streamer is inactive. As we want to gather feedback for channels that receive comments at a rate that is higher than the streamer could read new comments, we define a channel as "highly active" if it surpasses a rate of more than 60 comments per minute at least once during the recorded months. Approximately 16,600 channels fulfill this requirement. The highest recorded rate in these channels was nearly 11,000 comments per minute, the mean rate was approximately 180, and the median rate was 105 comments per minute. The top 10 most active channels by this metric are ddolking555 (max. 10933 comments/minute), twitch (max. 5602 comments/minute), geekandsundry (max. 4831 comments/minute), hanryang1125 (max. 3530 comments/minute), gotaga (max. 3376 comments/minute), yoda (max. 3137 comments/minute), zerator (max. 3129 comments/minute), riotgames (max. 2985 comments/minute), forsen (max. 2941 comments/minute), and kendinemuzisyen (max. 2911 comments/minute). From these channels, ddolking555 and hanryang1125 are Korean, gotaga and zerator are French, yoda is Portuguese, and kendinemuzisyen is Turkish.

Emote Usage. As shown above, emotes are a central part of Twitch comments. In the following, we analyze the use of the most popular emotes in some more detail.

Figure 6 shows the occurrences of these emotes. As with the general word counts, the number of usages per emote decreases drastically towards the end of the list. All of these 100 emotes combined make up approximately 4.77 % of all words in the dataset. Approximately 13.7 % of all comments contain at least one of these 100 emotes, which is stable across all recorded days (minimum: 12.35 %, maximum: 16.13 %, mean: 13.64 %, standard deviation: 0.47 %, median: 13.58 %). In top channels receiving the most comments during the recorded three months, the emotes that are used stay more or less the same over time. This shows that the fluctuation of emote usage is negligible over consecutive months, so the top 100 emotes are well-suited for longer-term analyses. However, in channels with fewer comments, emote usage shows high variance over time as well as between channels. This is due to the large number of users that often goes along with the

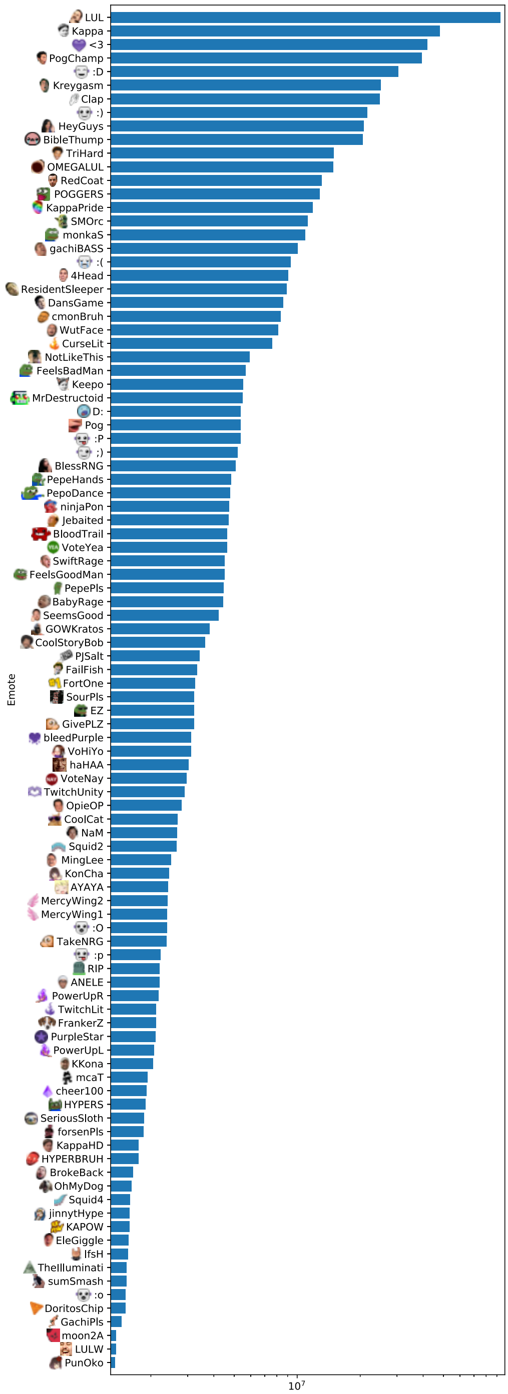

Fig. 6. Number of occurrences in the unlabeled dataset for the top 100 emotes.

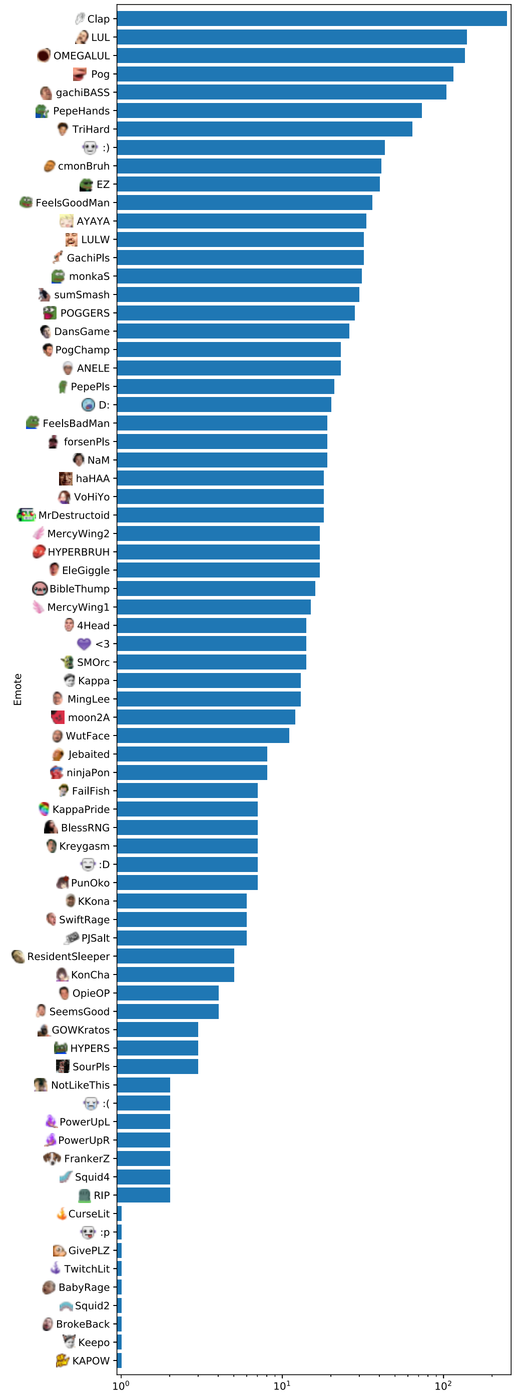

Fig. 7. Number of occurrences in the labeled dataset for the top 100 emotes obtained by analyzing the unlabeled dataset. Only 74 of 100 emotes are present in the labeled dataset.

number of comments a channel receives. The bigger the community of a channel, the more patterns emerge that are representative for the complete Twitch community, as personal linguistic and emote preferences do not have a big impact on the overall data.

4.2 Labeled Data Analysis

After analyzing the unlabeled data, this section provides some insights into the labeled dataset obtained by crowd sourcing (see Section 3.2). We first analyze the most frequent words for the sentiment classes and then investigate the frequency of emotes in this sample of the data.

Comment Length and Unique Words. The sampled dataset has a mean word count of approximately 5.5, a median word count of 2, and a standard deviation of approximately 8.75. The shortest comment is one word long, the longest has 84. A comment has on average approximately 4.14 unique words, with the median at 2 and a standard deviation of approximately 5.97. The maximum is at 63 unique words.

This shows that, regarding the mean comment length and number of unique words, the sampled test dataset is similar to the unlabeled dataset. Very long comments in the unlabeled dataset mostly contain very few unique words. Thus, they were selected with lower probability due to the sampling strategy explained in Section 3.2. Other long comments with more unique words were also selected with lower probability as we normalized the number of unique words with the comment's length.

Table 5. Most common words in comments given a sentiment polarity. In parentheses, the number of comments containing the corresponding word is given. Additionally, the percentage of the occurrences w.r.t. all comments with this sentiment is shown.

	Negative (404)		**Neutral** (748)		**Positive** (770)	
1	PepeHands	(48/11.9%)	Pog	(56/7.5%)	Clap	(102/13.2%)
2	@STREAMER	(23/5.7%)	@STREAMER	(22/2.9%)	OMEGALUL	(91/11.8%)
3	monkaS	(22/5.4%)	@USER	(18/2.4%)	LUL	(83/10.8%)
4	DansGame	(14/3.5%)	LULW	(16/2.1%)	gachiBASS	(48/6.2%)
5	BibleThump / FeelsWeirdMan	(12/3%)	LUL	(14/1.9%)	:)	(32/4.2%)

Most Frequent Words. Table 5 shows the top five words that are present in the labeled comments for each of the three sentiment polarities. These lists exclude common stop words. Similar to the findings of Section 4.1, most of these words are emotes. Emotes like PepeHands or DansGame are used more frequently in comments that are classified as negative, as they express sadness and disgust, respectively. Emotes like Clap and OMEGALUL are more often classified as positive, as they depict support/appreciation and laughter, respectively. These categorizations seem to be reasonable. Other emotes like LUL are present in multiple categories, showing that emotes cannot always be represented with a single sentiment score.

Mentioning the streamer or another user (here, mentions of the streamer are summed up as "@STREAMER" and mentions of other users are summed up as "@USER") seems to be more common for negative and neutral comments. Mentioning a user using the "@"-symbol notifies the mentioned user, which allows reactions and conversations between users in the chat.

Emote Usage. As we already extracted a list of the top 100 emotes found in the unlabeled dataset, we can now explore the usage of these emotes in the labeled dataset. Figure 7 depicts the number of occurrences in the relatively small labeled dataset for the extracted top 100 emotes, even though only 74 were present. The decay in emote use is similar to the one found in the unlabeled dataset, but the order of the emotes is different from the order in the unlabeled corpus. This may be due to certain emote preferences of the users of the chosen channels or to the sampling of the labeled dataset. We sampled comments with more unique words with a higher probability. As emotes are very popular on Twitch, this means that the sampled comments are more likely to have a more diverse set of emotes in it. This may be one reason why the percentage of comments containing at least one of the top 100 emotes increases to 47.92% in the labeled corpus.

5 SENTIMENT ANALYSIS ON TWITCH.TV

Sentiment analysis is a highly researched field in natural language processing that develops methods to estimate the sentiment of written text. It is useful to estimate the text's sentiment to automatically gather feedback for products and persons from large corpora of text, such as social media posts. In this work, we use Twitch.tv comments to automatically estimate the audience's sentiment throughout a stream to allow streamers to analyze their presentation and companies to improve their products.

The basic task in sentiment analysis is to classify a text as one of the classes "positive", "negative" and "neutral". Due to Twitch comments being fairly short and often not containing punctuation, we follow the structurally similar setting employed by the sentiment classification tasks on Twitter messages of [Rosenthal et al. 2017], which is predicting the sentiment of entire comments. Additionally, due to the limited amount of manually labeled training data presented in 3.2, we restrict the methods investigated in this paper to unsupervised and weakly supervised classification approaches.

While in other domains sentiment analysis is often required to achieve very high accuracy, as a single error can have grave influence on the overall result, we can afford to trade some accuracy for efficiency: Highly active Twitch streams often receive hundreds of comments per minute. Thus, getting the majority of the comments' sentiment right will still show the correct trends and enable streamers to draw valuable conclusions about which content is well-received by the audience.

As we have shown above, Twitch comments typically use a very different language compared to common English. Therefore, most standard sentiment analysis approaches based on common English are unsuitable for this domain. We will show this by applying a simple, yet powerful baseline to our task, which fails to correctly determine sentiment on our dataset. We then introduce our sentiment analysis methods, that heavily rely on the emotes that make up a large part of communication on Twitch, as shown in Section 4. The results from all methods are then given in Section 5.2.

5.1 Methods

In this section we introduce multiple unsupervised and weakly supervised sentiment analysis methods using the given sentiment lexica described above. These methods are of increasing complexity, ranging from lexicon based to neural network based approaches. We then evaluate the methods on our labeled dataset.

5.1.1 Baselines. To measure a performance increase of our methods compared to other methods, we include several baseline approaches.

Random Baseline. This very simple baseline consists of two possible strategies: (i) Sampling uniformly from the three possible sentiment labels for each comment and (ii) exploiting the knowledge about the distribution of the labeled dataset and then sampling randomly from this distribution.

Majority Baseline. The most common class in the evaluation dataset is "positive" with 40.06 %. This baseline always predicts the "positive" class.

VADER Baseline. As a more sophisticated baseline, we chose the sentiment analysis system that was proposed along with the VADER lexicon [Gilbert 2014] and is implemented in the Python NLTK module[22]. This module uses the VADER lexicon, as well as some rules to combine the word labels for predicting the overall sentiment of a text. Rules include intensification of all-caps words, dampening a word's sentiment if preceeded by "kind of", or negating the sentiment when a negation word is found.

VADER serves as a relatively strong baseline, as it was designed specifically for social media texts.

5.1.2 *Our Methods.*

We now introduce our methods that, besides the VADER lexicon, also take the other lexica introduced in Section 3.3 into account. Furthermore, we explore the differences between methods that utilize only average based sentiment labels and methods that take the distribution of sentiment ratings into account.

Preprocessing. The methods presented below require some amount of preprocessing of the comments' raw texts, which we describe in this section. The comments were lowercased and tokenized into words and punctuation while preserving emoticons, unicode emojis and capitalization of Twitch emotes. To standardize occurring words for our learning procedures, we replaced occurrences of urls with the tag "URL" and reduced characters occurring more than twice in succession in a word to two occurrences (for example, "loooove" is standardized to "loove").

When looking up tokens, in case of entries that are present in multiple lexica, the emote lexicon takes precedence over the emoji lexicon, which in turn supersedes VADER. We chose this prioritization as it represents how specialized the lexica are to the domain of Twitch comments.

Average Based Lexicon Approach. As a first specialized approach, we constructed a simple lexicon based classifier using the average based lexica L_{avg}, that provide a number $L_{avg}(t) \in [-1, 1]$ for a token t. After applying the preprocessing described above, each comment is represented as a sequence of tokens $T = (t_1, t_2, \ldots, t_n)$. We then create a new sequence $T^* = (t_i | t_i \in L_{avg})$ that consists only of the tokens that are present in at least one of the lexica. T^* is scored as follows:

$$\text{score}(T^*) := \begin{cases} \text{average}(L_{avg}(t) | t \in T^*) & \text{if } |T^*| > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the score of an entire comment is the average over all scores of tokens the lexica provide an entry for. This results in a continuous score between -1 and 1. To receive the final, discrete sentiment labels of *negative*, *neutral* or *positive*, thresholds were introduced as follows:

$$\text{sentiment}(T^*) := \begin{cases} \text{negative} & \text{if score}(T^*) < -0.33 \\ \text{neutral} & \text{if } -0.33 \le \text{score}(T^*) \le 0.33 \\ \text{positive} & \text{if } 0.33 < \text{score}(T^*). \end{cases}$$

[22] http://www.nltk.org/_modules/nltk/sentiment/vader.html#SentimentIntensityAnalyzer

Distribution Based Lexicon Approach. Our second approach is a generalization of the first one. As we have shown above, some emotes cannot be adequately represented by a single sentiment score, as they can be used in a positive or negative context. To be able to better exploit this knowledge, we replace the average based lexica L_{avg} from the previous approach with the distribution based lexica L_{dist}. Given the tokenized comment T, we again construct the sequence T^* of tokens that are present in the lexica L_{dist}. We now want to predict the correct class c for this list $T^* = (t_1, t_2, \dots, t_n)$ using $p(c|t_i)$ where $i \in 1, \dots, n$ and $c \in \{\text{negative}, \text{neutral}, \text{positive}\} =: C$. This can be done by assigning the most likely class c^* to T^* given $t_1, \dots, t_n$, that is $c^* = \operatorname{argmax}_{c \in C} p(c|t_1, \dots, t_n)$. The standard naive Bayes formula

$$c^* = \operatorname*{argmax}_{c \in C} p(c|t_1, \dots, t_n) = \operatorname*{argmax}_{c \in C} p(c) \prod_{i=1}^{n} p(t_i|c)$$

cannot be applied here as we want to classify in an unsupervised manner and do not have examples to infer $p(t_i|c)$ from. However, using Bayes' theorem and assuming conditional independence $p(t_1, \dots, t_n|c) = \prod_{i=1}^{n} p(t_i|c)$, it can be shown that

$$c^* = \operatorname*{argmax}_{c \in C} p(c|t_1, \dots, t_n) = \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} p(c|t_i) = \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} L_{dist}(t_i),$$

which only uses the sentiment distributions $p(c|t_i)$ given by our lexica L_{dist}. The proof is given in the supplemental material.

We used this fact to build a probabilistic classifier that computes c^* as its prediction, if any token in the comment is present in one of the lexica:

$$\text{sentiment}(T^*) := \begin{cases} c^* & \text{if } |T^*| > 0 \\ neutral & \text{otherwise.} \end{cases}$$

Sentence CNN. For both of the classifiers above, comments are labeled as "neutral" if they contain no signal tokens found in any of the three lexica. This can be caused by multiple reasons: For one, they may actually be neutral and therefore not contain any signals. On the other hand, many of the comments in Twitch contain orthographic mistakes or Twitch specific words which are not covered by any of the lexica but might still provide valuable information about the sentiment in a comment.

In order to also classify the remaining comments not covered by the lexicon approaches, we decided to use a neural network based classifier trained in a weakly supervised manner. To this end, we use the labels produced by our lexicon based approaches as training data for the network. Our intuition is that the network will be able to find the relation between a comment's sentiment and words that are not covered by the lexica, as well as being more robust to orthographical errors due to the embedding used as input. Frequent typos are given a similar embedding, and can therefore be evaluated correctly by the CNN, while a typo cannot be found in a lexicon.

As our neural network model, we use the Sentence CNN for sentiment analysis introduced in [Kim 2014]. This method consists of a Convolutional Neural Network (CNN) that takes as input sentences represented by a concatenation of word embeddings. These embeddings are then passed through multiple convolutions to extract relevant features before the final classification is done by a softmax-layer.

We use word2vec embeddings [Mikolov et al. 2013a] trained on the unlabeled corpus as input representation. To train the embeddings, we used the preprocessing described above on our unlabeled dataset and filtered all words that occurred fewer than 100 times. This means that some words were not available in the training phase of the network. Tokens that yielded no embedding were replaced by zero vectors.

We used the variant of the Sentence CNN described as cnn-nonstatic in [Kim 2014], which means that the pre-trained embeddings are fine-tuned along with the other network weights during training.

We kept to the task of not using manually labeled training instances as input data, by training the CNN in a weakly supervised manner: We trained the network by feeding the labels produced by our distribution based lexicon classifier as targets. This allowed us to produce weak labels for every comment in the unlabeled corpus, giving approximately three billion weakly labeled comments. The manually labeled corpus from Section 3.2 was then only used for evaluation purposes, making the entire process unsupervised.

The distribution based lexicon approach predicts a neutral label for every comment that has no token present in one of the used sentiment lexica. Since this is only a default assumption and not a label actually provided by the classifier, we decided to model these predictions as uncertain. Therefore, for any comment that does not contain any signal tokens from our lexica, we use a target distribution of 25% negative, 50% neutral, and 25% positive as target.

As this might lead to the network simply overfitting to this target distribution, we adapted a method proposed in [Go et al. 2009a]: removing signal tokens from the network's input. This forces the network to look for other words, phrases, and structures in the comment that correlate with its sentiment. In order to enable the network to rely on both signal tokens from the lexica and possible new clues, we replaced any signal token with a zero vector with a $50\,\%$ probability. We also used early stopping in order to prevent the Sentence CNN from overfitting. For this, we used approximately 20% of the training dataset as a validation set, for which the validation loss was calculated after every 500 batches consisting of 2816 training examples. If the validation loss did not improve during 5 consecutive validation iterations, we stopped and used the training state that produced the lowest loss.

To find the best hyper-parameters for the CNN, we added a random search. The hyper-parameters that we searched for were the filter count and dropout probability, following [Zhang and Wallace 2017]. After training about 30 different configurations, we selected the model with the lowest validation loss. This resulted in a Sentence CNN consisting of 182 filters and a dropout probability of 27% on the CNN layers during training.

5.2 Evaluation

Given the labeled dataset described in Section 3.2, we have a ground truth that can be used to measure the performance of our methods. The metrics for evaluating our results are the commonly used accuracy, macro recall and macro F1 score [Baeza-Yates et al. 1999].

Table 6 shows these three metrics for all methods on the labeled dataset. In the following paragraphs, we will provide some details about the performance of the classifiers.

Random Baseline. As a random procedure does not yield reproducible results, we report the expected measurements. Sampling uniformly from the three possible sentiment labels for each comment produces an expected accuracy of $33.3\,\%$, a macro recall of $33.3\,\%$ and a macro F1 score of $32.7\,\%$. By exploiting knowledge about the distribution of the labeled dataset and sampling randomly from this distribution, we can increase the expected accuracy to $35.6\,\%$ and macro F1 score to $33.3\,\%$, while the expected macro recall stays the same at $33.3\,\%$.

Majority Baseline. Always predicting the “positive” class, as it is the most frequent one in the dataset, leads to an accuracy of $40.1\,\%$, a macro recall of $33.3\,\%$, and a macro F1 score of $19.1\,\%$.

Table 6. Results for sentiment classification achieved by all methods

Method	accuracy	macro recall	macro F1 score
Random Baseline	33.3%	33.3%	32.7%
Random Baseline (sampling from target distribution)	35.6%	33.3%	33.3%
Majority Baseline	40.1%	33.3%	19.1%
VADER Baseline	43.0%	39.3%	34.0%
Average Based Lexicon Approach	61.8%	58.9%	60.5%
Distribution Based Lexicon Approach	62.8%	60.5%	61.7%
Sentence CNN	63.8%	61.4%	62.6%

VADER. Even though VADER is specifically designed for dealing with social media texts, the macro F1 score obtained by this method is 34.0 %, which is only a small increase in contrast to randomly selecting labels. This is due to the fact that the language used on Twitch is very different from common English (see Section 3.1) and even from common social media language. The macro recall and accuracy, however, increase to 39.3 % and 43.0 %, respectively.

Average Based Lexicon Approach. Our simplest approach based on multiple sentiment lexica yields an accuracy of 61.8%, macro recall of 58.9 % and a macro F1 score of 60.5 %. 65.2 % of comments in our evaluation data set contained tokens found in our lexica and were therefore labeled by the classifier. The other 34.8 % were assigned the "neutral" label by default. The large improvement over the baselines presented above shows that incorporating sentiment lexica for emoji and emotes can provide reasonable accuracy even with a rather simple classifier.

Distribution Based Lexicon Approach. This classifier using distribution based lexica achieves an accuracy of 62.8%, a macro recall of 60.5%, and a macro F1 score of 61.7 %, which is an improvement to the previous approach. As above, 65.2 % of the comments in the dataset had tokens found in the lexica, the remaining comments were labeled as neutral by default.

Sentence CNN. Weakly supervised training of the Sentence CNN on the labels produced by the distributional classifier obtained an accuracy of 63.8 %, a macro recall of 61.4 % and a macro F1 score of 62.6 %. This result improves the distribution based lexicon approach, even though the weak-labels were produced by the lexicon based method. We will provide some analysis of the reasons for this improvement in the following section.

6 ANALYSIS

We have shown that our proposed methods outperform baselines by a large margin. In the following, we provide some analysis of the methods and their results and compare them with each other. We also analyze the importance of our features (sentiment lexica and word vectors) more thoroughly and show that emotes have a big impact on the performance of our methods. We also show that word embeddings trained on the Twitch dataset contain semantic relations that can be uncovered using vector calculations.

Table 7. Results of both lexical approaches using different combinations of lexica. Best results are written in bold.

	accuracy Average/Distribution	**macro recall** Average/Distribution	**macro F1 score** Average/Distribution	% of comments containing signal tokens
Emoji	39.5 % / 39.8 %	34.0 % / 34.3 %	21.1 % / 21.4 %	3.8 %
VADER	48.3 % / 45.9 %	45.1 % / 43.1 %	42.1 % / 38.1 %	26.6 %
Emoji + VADER	48.5 % / 46.4 %	45.3 % / 43.6 %	42.7 % / 39.2 %	29.1 %
Emote	58.9 % / 59.7 %	54.3 % / 55.2 %	55.1 % / 56.0 %	48.0 %
Emote + Emoji	58.9 % / 60.3 %	54.4 % / 55.8 %	55.3 % / 56.7 %	50.6 %
Emote + VADER	61.8 % / 62.4 %	58.8 % / 60.2 %	60.4 % / 61.3 %	63.4 %
Emote + Emoji + VADER	61.8 % / 62.8 %	58.9 % / 60.5 %	60.5 % / 61.7 %	65.2 %

6.1 Ablation Study: Emotes Matter!

In order to validate our assumption that emotes have major influence on the sentiment of Twitch comments, we conducted an ablation study for our two lexicon based classifiers, investigating the influence of different lexica. We find that both approaches profit strongly from the inclusion of emotes. Table 7 shows the results for all combinations of the three lexica. Along with the measures accuracy, macro recall, and macro F1 score, the table shows the percentage of comments with at least one token found in the lexicon. The emoji lexicon does not improve the classification performance by much, but increases the amount of comments that are not simply assigned a default "neutral" label by two percentage points. It can be seen that all lexica are relevant to the classification, while the emote lexicon has the single largest influence. Also note that the emote lexicon covers more comments than any other lexicon. These findings are in line with our expectation that emotes are crucial for the understanding of comments on Twitch as well as our analysis of the dataset in Section 3.

6.2 Comparison of Approaches: Complexity Matters!

In addition to the ablation study presented above, we analyzed the differences between the predictions our classifiers make in order to enable a better understanding of their relative performance. Despite similar numeric results in the average and distribution based approach and a Spearman correlation coefficient of 0.88, there are a number of cases where the approaches classify comments differently. In fact, both approaches are significantly different from each other with a significance level of 1 %, based on the Randomized Matched-Pair Test from [Yeh 2000] (p-value for F1 score: 1.9×10^{-6}). Using the same test, comparing the CNN and distribution based approach also shows significant difference with a p-value for F1 score of 2.9×10^{-6}.

As previously mentioned, approximately 35% of comments in the evaluation dataset did not contain tokens present in the lexica. These comments were assigned the "neutral" label per default by the lexicon based approaches. When comparing the results of all three classifiers, it is noticeable that in contrast to our expectations, the CNN did not improve the classification of these comments. Almost all of these 669 comments (that is, 35% of the evaluation data set) were also classified as neutral by the CNN.

The overlap of correctly classified comments is highest with CNN and the distribution based classifier at 62% correctly classified comments. This means that for 62% of all comments, which were classified correctly, the CNN and the distribution based classifier predicted exactly the same label. This is most likely due to the CNN being trained using labels predicted by the distribution based classifier. Both the overlap of CNN and average based lexicon approach as well as average based and distribution based approach only contain 58% correctly classified comments.

Table 8. Distribution of classified comments of all three approaches and the original evaluation data set

Classifier	Negative	Neutral	Positive
True Sentiment	404	748	770
Average Based Classifier	237	1027	658
Distribution Based Classifier	290	971	661
Sentence CNN	281	962	679

As seen in Table 8, the lower amount of comments which are classified as neutral by the CNN seems to be the largest influence for the improved performance over the average based and distribution based approaches.

Table 9. Amount of comments labeled as negative/neutral/positive by the classifiers in comparison to the true sentiment. Excluding comments with default neutral sentiment due to undetected tokens.

True Sentiment \ Estimated Sentiment	Average Based Lexicon Approach			Distribution Based Lexicon Approach			Sentence CNN		
	neg.	neu.	pos.	neg.	neu.	pos.	neg.	neu.	pos.
negative (312)	**174**	112	26	**195**	80	37	**193**	83	36
neutral (320)	49	112	**159**	71	104	**145**	63	111	**146**
positive (621)	14	134	**473**	24	118	**479**	25	100	**496**

Table 9 compares the sentiment predicted by the average based lexicon classifier in comparison to the true sentiment. The table shows that, with the exception of true neutral comments, the classifiers show a clear tendency to correctly classify negative and positive comments. Only very few comments have completely contrary sentiment where the classifiers predict negative sentiment for a comment labeled as positive by the crowd workers or vice versa.

6.3 Analyzing Twitch Embeddings: Domain Matters!

The Sentence CNN described in Section 5.1.2 uses word2vec embeddings to encode words. Word embeddings are known to group semantically similar words to similar vectors and enable vector calculations that show semantic relationships between words and their corresponding embeddings [Levy and Goldberg 2014]. Given the language properties of Twitch comments described earlier, it is not clear that word2vec trained on Twitch captures the word semantics as was shown on other corpora. In this section, we show that Twitch embeddings encode both general semantic information (like the well-known “king - man + woman = queen” example) as well as more domain specific jargon. Thus, they allow us to further inspect the language domain of Twitch and analyze the semantic information encoded in emotes, as our word2vec model also contains embeddings for emotes. We are able to perform simple vector calculations on emote embeddings and transfer standard semantic evaluation tasks to the domain of emotes. From this, we can derive that emotes do indeed carry a semantic component.

In the following, we perform standard tasks that can be solved by querying the embedding. In the tasks, we query both common English words as well as gaming-related words and Twitch specific emotes to show that our embedding is able to capture both common and more specialized relations in the language used on Twitch. We qualitatively selected the example queries in order to present the specialties of our embedding. We payed attention to picking diverse examples in terms of emotion and domain, that is general and gaming related queries as well as positively and negatively perceived emotes. We then also compare the performance of the Sentence CNN using Twitch embeddings with the performance of the model using two pre-trained embeddings based on other corpora.

Task 1: Detection of the Odd Word. Given a list of words, the model determines the one that does not fit the other words, that is, the word from the list that has a vector farthest away from the mean of all vectors. As Table 10 shows, the embedding can identify the correct word for both the general and the domain specific case.

Table 10. Task 1: Detection of the odd word from a list of words. Detected outliers are underlined.

Domain	List of words	Explanation
General	breakfast, <u>cereal</u>, dinner, lunch	Food in a list of meals
	apple, <u>cucumber</u>, peach	A vegetable in a list of fruits
Gaming	youtube, twitch, <u>instagram</u>	An image centric social network in contrast to video and streaming centric social networks
	fortnite, <u>overwatch</u>, pubg	Overwatch is not a game in the “Battle Royale” genre

Task 2: Words that Fit in a Given Context. Given a list of cue words, the model will find words that fit into that context, that is, get the words with the smallest embedding distance to the mean of the cue words. Again, we evaluate general and domain specific queries. The results are shown in Table 11.

As word vectors are dependent on the context the words are used in, words that are used in multiple contexts may have representations that are “averages” of the different meanings. For example, Table 11 shows that the representation of “Friday” is farther away from the other weekday representations. This is most likely due to the alternative use of “Friday” in the video game title “Friday, the 13th”, which is a popular game often streamed on Twitch.

The examples show that the domain of the training data is affecting the embedding representation due to the different context words are used in. The game centric and game play related community language results in some different query results and allows for domain specific queries that are not possible to perform on other word corpora, as shown in Table 11. Embeddings trained on the Google News corpus do not yield answers that reflect the opinion or language of the Twitch domain.

Task 3: Word Relations. The possibility to do “semantic calculations” with word embedding vectors is one of the most impressive properties of word embeddings. One of the most famous examples is provided in [Mikolov et al. 2013b]: In embedding space, “man” relates to “woman” as

Table 11. Task 2: Finding words that fit a context given by a list of cue words. In comparison, the responses from a word2vec embedding trained on the Google News corpus is also given.

List of cue words	**Fitting words** (Twitch)	**Explanation**	**Fitting words** (Google News)[a]
monday, tuesday, wednesday	thursday, saturday, sunday, friday	Days of the week	thursday, friday, saturday, sunday
battlefield, cod	halo, battlefront, titanfall	Shooter games	Cod, battlefields, herring
witcher, wow, skyrim	bloodborne, fallout, morrowind	Role-playing games	— (“skyrim” not in vocabulary)

[a]Pre-trained embeddings taken from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

“king” to “queen” [Levy and Goldberg 2014]. This kind of query can be performed for other word pairs as well. Table 12 shows some queries we performed on our embedding model, including the previously mentioned prime example. Again, embeddings trained on Google News were queried with the same word pairs to show the difference of the Twitch comment language and common English.

Task 4: Intensification of Emotes. As mentioned above, our embedding model also includes vector representations of emotes. We therefore can also query the embedding using emotes and their relations. Word vectors are derived from the context they are used in. If we can show that embeddings of these emotes result in sane query results as well, we have shown that certain emotes are used in certain contexts, thus having a semantic component. As emotes are very popular on Twitch, exploiting this semantic component like any other word in the English vocabulary can have a big influence on the performance of sentiment analysis methods, as shown in our experiments.

As in the previous task, we query the embedding to get the word, in this case the emote, that most closely resembles the given relation. In these examples, we use the relation of OMEGALUL , which is an exaggeration of LUL , as shown in Table 12. OMEGALUL originates from LUL , but the mouth is warped to fill the largest part of the image. We use the relation of these two emotes to search for exaggerations and intensifications of other emotes. Table 13 depicts some intensifications of some of the most popular emotes that were found using the embedding.

Overall, we have shown that our embeddings, which were calculated only based on Twitch comments, provide good results on Twitch specific queries. Our embeddings enable these queries while at the same time being able to model general word relations. The results for the general queries don’t always match with the answers given by an embedding trained on the Google News corpus, but instead model the language in the Twitch chat. This is of course beneficial for our sentiment classification task.

Table 12. Task 3: Finding word pairs that have the same relation as another word pair. We always set the first three words (A, B, and C) to find the related forth word (X). In comparison, we queried embeddings trained on the Google News corpus.

A relates to B as C to X					
A	**B**	**C**	**X (Twitch)**	**Explanation**	**X (Google News)**[a]
man	woman	king	queen, princess, goddess	Prime example of word embedding calculations	queen, monarch, princess
man	woman	greekgodx	kaceytron, kwehzy	Finding the streamer greekgodx's female counterpart	— ("greekgodx" not in vocabulary)
hero	loser	gamer	gaymer	Homophobic insults	gamer, losers, gamers
rockstar	gta	blizzard	overwatch	Game development companies and their products	snowstorm, blizzards
LUL	OMEGALUL	large	huge, big, massive, gigantic	Intensifications of adjectives using emotes	— ("OMEGALUL" not in vocabulary)

[a]Pre-trained embeddings taken from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

In order to quantify this difference between embeddings, we also evaluated Sentence CNN on embeddings trained on the Google News corpus[23] as well as Twitter [Godin et al. 2015]. For training, we used the same hyper-parameters as described in Section 5.1. Table 14 shows the Sentence CNN performance using the respective embedding. The Google News embedding ranks worst, while the more social interaction focused Twitter embedding fares better. However, our Twitch embedding achieves the best performance, with a difference of more than ten percentage points. We suspect that this is due to the lack of emote representations in the Google News and Twitter embeddings.

[23]Taken from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

Table 13. Task 4: Finding emote pairs that have the same relation as LUL to OMEGALUL. Emote Y is provided by the embedding.

LUL **relates to** OMEGALUL **as X to Y**		
X	**Y**	**Explanation**
FeelsGoodMan	FeelsAmazingMan	Approval/satisfaction intensifies to amazement
FeelsBadMan	PepeHands	Sadness is intensified by crying
EZ	POGGERS	Extraordinary moves and moments in the (game) stream
cmonBruh	HYPERBRUH	An emote that is mostly used if the streamer's commentary can be interpreted as racist intensifies to an emote that is used in situations of clear racism.
WutFace	(puke)	Puking often follows disgust
4Head	4House	Intensifications in the emote text representations
4House	4Mansion	

Table 14. Results for sentiment classification achieved using different embeddings.

Method	accuracy	macro recall	macro F1 score
Twitch embedding	63.8%	61.4%	62.6%
Twitter embedding	53.3%	50.2%	50.6%
Google News embedding	53.3%	48.7%	48.3%

7 CASE STUDIES

In the previous sections, we have introduced several methods to conduct sentiment analysis on comments made by users on Twitch.tv and have shown that they are able to recognize the sentiment encoded in the comments with reasonable accuracy. In this section, we show that the accuracy achieved by our methods is high enough to provide streamers with feedback regarding their streams. To this end, we conduct two case studies, analyzing events that have been live streamed on Twitch.tv. The first of these events is Nvidia's presentation of the GeForce RTX family of graphics cards. The second event is the keynote from BlizzCon 2018, where, among others, the new game "Diablo Immortal" was introduced. We show the effectiveness of our proposed methods by using them to perform sentiment analysis on the comments made during these events, forming sentiment trajectories over time and analyzing how well the peaks in these trajectories correspond to events in the stream.

For both events, we queried all comments made during the presentations in the respective company's official channel. We then used the Sentence CNN presented in Section 5.1.2 to perform sentiment classification on these comments. We smoothed the resulting sentiment trajectory by applying a sliding window average over 5000 comments.

7.1 Nvidia RTX Unveiling

The first event we analyze is the presentation of the Nvidia RTX family on August 20th 2018. In this presentation, which was live streamed on Nvidia's official Twitch.tv channel[24], Nvidia introduced their latest GPU generation as well as presented some games taking advantage of the new architecture.

Applying the procedure described above for this presentation leads to the sentiment trajectory shown in Figure 8.

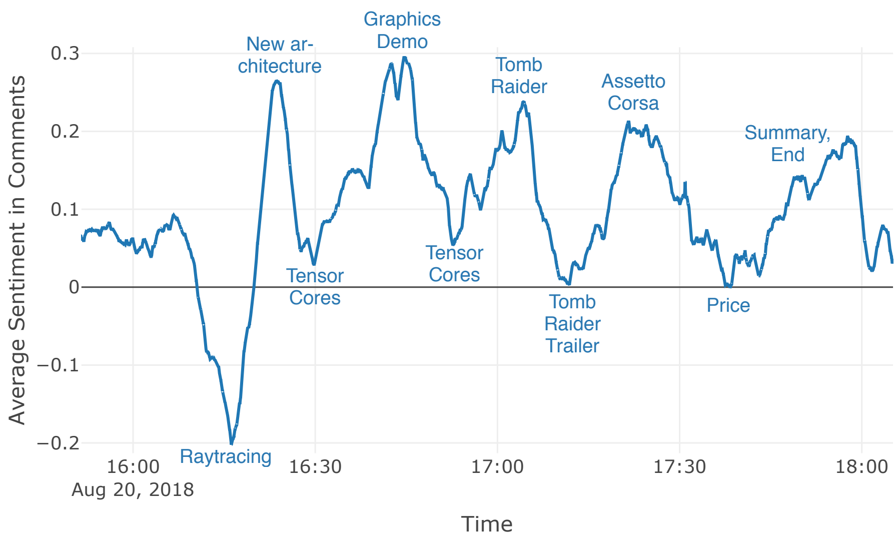

Fig. 8. Sentiment trajectory of the Nvidia RTX 2080 presentation

In order to analyze the viewers' sentiment towards specific events in the stream, we mapped the peaks in the trajectory to a video recording of the presentation[25].

Overall, the peaks correlate very well to events in the stream and the sentiment detected by our classifier is in line with the expected reaction from a gaming audience. Figure 10a shows the specific time stamps of the sentiment peaks and the topic of the presentation at that time. Generally, we notice that the sentiment is more positive for gaming-related topics (the announcement of the new cards, the graphics demo, Tomb Raider, Assetto Corsa) and negative for technical details and specifically machine learning-related topics. This is in line with the general assumption that the audience on Twitch.tv has a very high affinity to gaming-related topics. The only exception to this rule is the screening of a trailer for Shadow of the Tomb Raider. We assume that this screening received more negative comments than other gaming-related topics because the content of the trailer was already known before the event.[26] The trailer also did not contain any thrilling new details and could therefore be perceived by the audience as boring.

[24]https://www.twitch.tv/nvidia
[25]https://www.youtube.com/watch?v=Mrixi27G9yM
[26]The first trailer for the game had been released in april 2018, 4 months prior to the Nvidia presentation.

Fig. 9. Times of sentiment peaks in the case study events and corresponding topics

(a) Nvidia RTX presentation

Time	Topic
16:16	Technical details about Raytracing
16:24	Announcement of new architecture
16:30	Technical details about generating missing pixels using AI (Tensor Core)
16:44	Graphics Demo Video
16:52	Tensor Cores, Deep Learning Super Sample (DLSS), Convolutional Auto Encoder
17:04	Announcement: Shadow of the Tomb Raider
17:11	Tomb Raider Trailer
17:21	Frame analysis of Assetto Corsa
17:38	Presentation of actual hardware with price tag
17:42	Summary, End

(b) BlizzCon 2018 keynote

Time	Topic
17:45	Discussion streamed before the keynote
18:20	Report on the Pink Mercy skin campaign
18:45	Long talk in advance of Warcaft video
18:57	Overwatch Cinematic
19:20	Diablo Immortal announcement

7.2 BlizzCon Keynote

The target event of our second case study is the 2018 edition of the annual convention "BlizzCon" held by video game company Blizzard on November 2nd-3rd of 2018. This convention is dedicated to all franchises published by Blizzard, as for example Starcraft, Diablo, and Warcraft. Each BlizzCon is prefaced by a presentation, where new games and new content for existing games is announced. This year's opening presentation was live streamed publicly on Blizzard's Twitch channel[27].

It contained an announcement that was especially controversially received by fans, that is, the reveal of a mobile game called "Diablo Immortal". Previous to BlizzCon, fans were anticipating a sequel to the PC game Diablo III. The release of a Diablo game for smartphones instead of PC was heavily critizised [Polygon 2018] and even led to the presenter on the stage being booed. This very strong reaction makes the event a suitable benchmark for our methods, as the sentiment should drop significantly at the point of this reveal.

Analyzing the sentiment in the comments made during the presentation leads to the trajectory shown in Figure 11, which can be mapped to the events shown in Figure 10b. The announcement of Diablo Immortal took place at the end of the keynote at about 19:20. Looking at this time in the sentiment trajectory, we can indeed see a slight rise in the commenters' sentiment when the topic of Diablo is first mentioned, which is then followed by an extremely steep drop as soon as the audience becomes aware that the game will be released for smartphones only. Analyzing the other peaks in the trajectory, we find that the discussion of the success of a campaign supporting breast cancer research by donating revenue from a skin in the game Overwatch[28] (about 18:20) as well as a new Overwatch cinematic (about 18:57) have been very well received, while the drops

[27] https://www.twitch.tv/blizzard
[28] https://playoverwatch.com/en-us/news/21931801

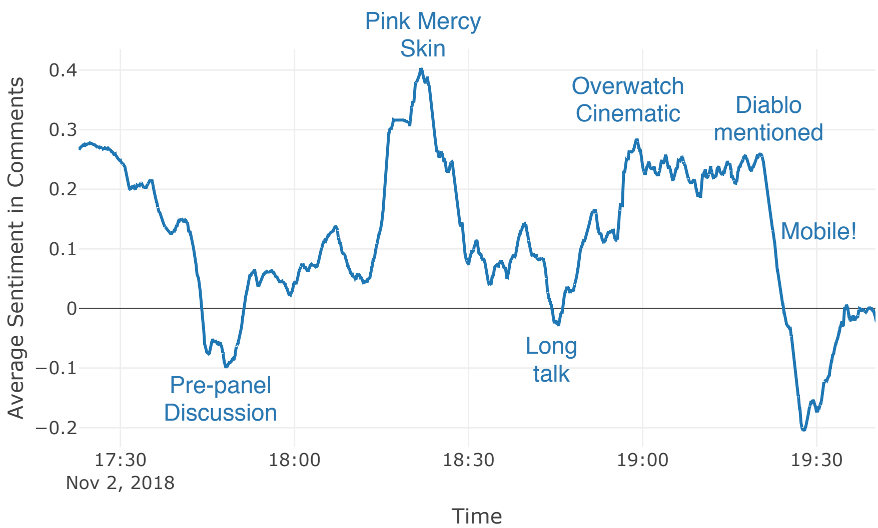

Fig. 11. Sentiment trajectory of the keynote presentation of BlizzCon 2018

at around 17:45 and 18:45 correspond to a discussion streamed before the actual presentation and a longer section of talk in advance of a Warcraft trailer, respectively. Both of these parts were apparently perceived as boring by the audience, as evidenced by the frequent occurrence of the emote  ResidentSleeper in comments around these times.

8 DISCUSSION

In this work, we have presented methods that are able to reliably estimate the sentiment of Twitch comments, which in turn allows streamers to visualize trends in the audience's sentiment to get feedback for their product or stream and perhaps adapt their presentation accordingly. This section will discuss our findings in some more detail.

The basic assumption of this paper was that, due to the very specific language of Twitch comments, generic sentiment analysis approaches would fail to provide satisfactory classifications on Twitch data. We also hypothesized that using emotes could be a way to overcome the challenge posed by this language, as they make up a large part of Twitch comments. Our experiments show that both of these assumptions are correct: The VADER baseline, even though it is designed for social media texts, cannot capture the sentiment expressed in Twitch comments. Our methods however, which include sentiment information about emotes in addition to words and emojis, are able to detect sentiment with reasonable accuracy. Our ablation study has shown that this is indeed mostly due to the emote lexicon.

A common shortcoming of lexicon based classifiers is their inability to deal with spelling errors or, more, generally, words not contained in the lexica they are based on. We proposed to use a Convolutional Neural Network based on word embeddings to enable generalization to unknown words. Our analysis shows that, while the CNN does indeed perform better than the lexicon based classifiers, this improvement is not due to the generalization we had hoped for. This could be due to the network overfitting to the target distribution given by our distribution based lexicon classifier. We had hoped that marking the default neutral classification for comments that do not contain words in our lexica as uncertain by representing it as 25 % positive, 50 % neutral and 25 % negative would be enough to prevent this, however this does not seem to be the case. Exploring other methods to enforce better generalization is an interesting topic for future work. Possible approaches include providing a target distribution closer to the uniform distribution, deleting uncertain training examples with a given chance, or modifying the learning rate of the neural network to be lower when the label is uncertain.

Despite the better results, the CNN requires time-consuming training and hyper-parameter optimization as well as rather large amounts of storage space for the embeddings and weights compared to the lexicon based approaches. While this does not pose a significant problem for most applications, it could be relevant for real-time use. Our lexicon based classifiers could easily be integrated into a browser plugin to provide streamers with real-time information about their audience's sentiment and enable them to adjust their stream accordingly. On the other hand, the slightly higher accuracy of the CNN could be useful for off-line analysis of comments after events as those presented in our case studies.

In Section 3.2, we mentioned that we had no way of ensuring the familiarity of the workers in the crowdsourcing campaign with the language on Twitch. While the Kappa-score shows an inter-annotator agreement that is comparable to similar campaigns on Twitter data, we wanted to analyze the quality of the annotations at least by spot checks. To this end, we gave the resulting labeled dataset to two experts in the Twitch domain. Both agreed that only a negligible amount of comments was rated completely wrong by the workers due to misunderstanding the included emotes. Therefore, we are certain that the resulting data was sanely labeled for the majority of examples. However, there are some emotes that seem to be frequently misunderstood by crowd workers. One example for this is the emote  Pog , which resembles only the mouth of the popular emote  PogChamp . Hence, it is a positive emote, as also indicated by our emote lexicon. In the labeled dataset, many of the examples containing this emote, however, were labeled as neutral. This may be due to no direct visual cue that this emote is positive, while the sentiment of other emotes can be assessed by just looking at them. For example, the emote  resembles a smiling face, thus directly showing a positive sentiment. Additionally, as emotes are influenced by internet trends, their meaning might not be obvious without knowledge about their background story. Other emotes can be used in multiple situations, such as  . This emote shows surprise regardless of the sentiment. Thus, short messages without other cue words are not easy to classify. Annotators therefore might disagree in the connotation of a message and the corresponding emote. While such misunderstandings might lead to deviations of a few percentage points for the classifiers' scores, our case studies still show that the classifications produced by our methods are well-suited for the analysis of viewers' reactions to events in the stream. In fact, when removing all examples from the labeled evaluation dataset that contain the emote  Pog , the Sentence CNN improves its accuracy to 65.6 %, its macro recall to 63.3 % and its macro F1 score to 64.3 %, which are approximately two percentage points per metric. The labeled dataset therefore potentially underestimates the performance of our methods as the workers tend to conservatively choose neutral instead of the correct positive sentiment. Future work might include building a more representative labeled dataset that was labeled by domain experts and evaluated in the given Twitch stream context. This, however, is a very costly and time-consuming task. Creating a labeled dataset that captures the sentiment trends of stream comments may be an easier and more cost-effective alternative, which could be used to quantitatively evaluate the case studies we conducted.

9 RELATED WORK

Sentiment analysis is a widely researched application area of machine learning. Next to popular usage areas containing datasets of Amazon product reviews [McAuley et al. 2015] and IMDB movie reviews [Maas et al. 2011], lots of studies have also challenged more difficult domains such as the short, orthographically inconsistent messages found on Twitter [Nakov et al. 2016, 2013; Rosenthal et al. 2017, 2015, 2014]. Research in this area also entails the use of emojis for gaining insights into the sentiment of a message [Kralj Novak et al. 2015], while there are also openly available resources for sentiment classification specifically geared towards social media texts, such as the VADER Valence Aware Dictionary for sEntiment Reasoning of [Gilbert 2014]. Additionally, there exist labeled datasets such as the Sentiment140 Twitter dataset of [Go et al. 2009b] utilizing text emoticons at the end of messages to generate a large amount of so called weakly labeled messages for use in supervised training environments.

Next to investigations on the sentiment of texts, finding task appropriate text embeddings to allow the application of classifiers such as neural networks has been a research focus in recent years. While embeddings containing syntactic similarities of words can be easily generated through methods such as one-hot encoding, [Tang et al. 2014] generate embeddings that express the semantic similarity of words on a corpus of Twitter messages. Next to the most commonly used semantic word embeddings word2vec [Mikolov et al. 2013a] and FastText [Bojanowski et al. 2017], there also exists the publicly available emoji2vec embedding of [Eisner et al. 2016] that tries to catch the semantic relation of unicode emojis.

The streaming platform Twitch itself has also gathered some research interest over the years. For a general overview of Twitch and its user communities, we refer readers to [Smith et al. 2013]. While [Kaytoue et al. 2012] analyzes viewer numbers and proves aspects such as the impact of tournaments and video game releases, [Nascimento et al. 2014] conducts a more indepth research on behavioral patterns of audiences, such as channel switching and channel surfing. There also exist studies in the area of viewer sentiment, with [Löffler et al. 2017] investigating the impact of background color on the perceived sentiment of chat comments. [Barbieri et al. 2017] researched the process of removing ending Twitch emotes from comments and predicting the removed emotes with Bidirectional Long-Short-Term-Memory Neural Networks. Predicting the overall sentiment of individual comments on Twitch, however, is a novel contribution of this paper.

10 CONCLUSION

In this work, we have presented methods that are able to reliably detect sentiment in comments on Twitch.tv and have shown that these methods can be used to analyze the general mood in the audience over the course of a stream. To this end, we have introduced a large unlabeled dataset of Twitch comments and provided a subset of this data manually labeled with sentiment information. We have also conducted in-depth analyses of the language used in these comments, showing the enormous importance of emotes for their understanding. Our methods are overall unsupervised and do not require manually labeled data for training. Our datasets and code are publicly available at https://github.com/konstantinkobs/emote-controlled.

The methods we have developed can be used by companies and streamers to optimize their streams for the best possible reaction among their audience, or to gather feedback about their products and presentations. They can easily be integrated into browser plugins for real-time feedback.

We see opportunities for future work in the improvements of our methods on the one hand, as outlined in Section 8. On the other hand, we have laid a foundation for the analysis of the language used in Twitch. Analyzing, for example, the differences in language between esport streams and standard streams as well as the evolution of Twitch language over time is now possible. Further exploring this language in the context of the culture exhibited by Twitch users is an exciting possibility for cooperation with researchers from sociology and political sciences. Enriching the word embeddings trained on Twitch comments by combining them with other common embeddings, such as word2vec trained on the Google News Corpus, could help models to consider different meanings of words. For example, the word "duty" is mostly used because of the game "Call of Duty" in Twitch comments, while another embedding can add the more common meaning of this word. The exploration of such word embedding enhancements can be considered as future work.

In terms of evaluation, new forms of evaluation methods might be of interest, such as the already mentioned dataset consisting of sentiment trends for different kinds of streams. More case studies on different kinds of streams might help to further qualitatively investigate the effectiveness of our methods.

ACKNOWLEDGMENTS

This work was partially done in a Computer Science Master's project at the Julius-Maximilians-University of Würzburg. While some of the students were involved in the writing of this paper, other students also were part of the project. These are: Tobias Greiner, Kevin Makowski, Julian Walter, and Nils Wehner. We also want to thank Dr. Toni Wagner for providing the Twitch data crawler. Special thanks also to Jonathan Liebig[29] and Matteo Ricciardi[30] for help with getting participants for the emote sentiment survey and to Nvidia Corporation for their support in form of the Academic GPU grant program.

REFERENCES

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.

Francesco Barbieri, Luis Espinosa Anke, Miguel Ballesteros, Juan Soler, and Horacio Saggion. 2017. Towards the Understanding of Gaming Audiences by Modeling Twitch Emotes., Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (Eds.). Association for Computational Linguistics, 11–20. http://dblp.uni-trier.de/db/conf/aclnut/aclnut2017.html#BarbieriABSS17

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on Italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 100–107.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. 2007. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 1–14.

Xu Cheng, Cameron Dale, and Jiangchuan Liu. 2008. Statistics and social network of youtube videos. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*. IEEE, 229–238.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. emoji2vec: Learning Emoji Representations from their Description.. In *SocialNLP@EMNLP*, Lun-Wei Ku, Jane Yung jen Hsu, and Cheng-Te Li (Eds.). Association for Computational Linguistics, 48–54. http://dblp.uni-trier.de/db/conf/acl-socialnlp/acl-socialnlp2016.html#EisnerRABR16

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.

CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Alec Go, Richa Bhayani, and Lei Huang. 2009a. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1, 12 (2009).

Alec Go, Richa Bhayani, and Lei Huang. 2009b. Twitter Sentiment Classification using Distant Supervision. *Processing* (2009), 1–6. http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf

[29] https://twitter.com/JJLiebig

[30] https://twitter.com/mentalmimicry

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations. In *Proceedings of the Workshop on Noisy User-generated Text*. 146–153.

Martin J Halvey and Mark T Keane. 2007. Exploring social dynamics in online media sharing. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 1273–1274.

HS Heaps. 1978. Information Retrieval-Computational Aspects Academic Press. *New York* (1978).

Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira Jr., and Chedy Raïssi. 2012. Watch me playing, i am a professional: a first study on video game live streaming. In *WWW (Companion Volume)*. ACM, 1181–1188. http://dblp.uni-trier.de/db/conf/www/www2012c.html#KaytoueSCMR12

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1746–1751.

Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of emojis. *PLoS ONE* 10, 12 (2015), e0144296. http://dx.doi.org/10.1371/journal.pone.0144296

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*. 171–180.

Ruokuang Lin, Qianli D. Y. Ma, and Chunhua Bian. 2014. Scaling laws in human speech, decreasing emergence of new words and a generalized model. arXiv:cs.CL/1412.4846

Diana Löffler, Lennart Giron, and Jörn Hurtienne. 2017. Night Mode, Dark Thoughts: Background Color Influences the Perceived Sentiment of Chat Messages.. In *INTERACT (2) (Lecture Notes in Computer Science)*, Vol. 10514. Springer, 184–201. http://dblp.uni-trier.de/db/conf/interact/interact2017-2.html#LofflerGH17

Vittorio Loreto, Vito DP Servedio, Steven H Strogatz, and Francesca Tria. 2016. Dynamics on expanding spaces: modeling the emergence of novelties. In *Creativity and universality in language*. Springer, 59–83.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 142–150. http://dl.acm.org/citation.cfm?id=2002472.2002491

Julian J. McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring Networks of Substitutable and Complementary Products. *CoRR* abs/1506.08839 (2015). http://dblp.uni-trier.de/db/journals/corr/corr1506.html#McAuleyPL15

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic Regularities in Continuous Space Word Representations.. In *HLT-NAACL*. 746–751.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*. 1–18.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in Twitter. 2 (2013), 312–320.

Sascha Narr, Michael Hulfenhaus, and Sahin Albayrak. 2012. Language-independent twitter sentiment analysis. *Knowledge discovery and machine learning (KDML), LWA* (2012), 12–14.

Gustavo Nascimento, Manoel Ribeiro, Loïc Cerf, Natália Cesário, Mehdi Kaytoue, Chedy Raïssi, Thiago Vasconcelos, and Wagner Meira. 2014. Modeling and analyzing the video game live-streaming community. In *Web Congress (LA-WEB), 2014 9th Latin American*. IEEE, 1–9.

Polygon. 2018. Diablo: Immortal broke the unspoken rules of Blizzard, and BlizzCon. https://www.polygon.com/2018/11/5/18064290/blizzard-diablo-immortal-reaction-explainer-blizzcon Accessed 2018-11-20.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 502–518.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 451–463.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 73–80. http://www.aclweb.org/anthology/S14-2009

Herbert A Simon. 1955. On a class of skew distribution functions. *Biometrika* 42, 3/4 (1955), 425–440.

Thomas Smith, Marianna Obrist, and Peter C. Wright. 2013. Live-streaming changes the (video) game. In *EuroITV*, Paolo Paolini, Paolo Cremonesi, and George Lekakos (Eds.). ACM, 131–138. http://dblp.uni-trier.de/db/conf/euroitv/euroitv2013.html#SmithOW13

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification.. In *ACL (1)*. The Association for Computer Linguistics, 1555–1565. http://dblp.

uni-trier.de/db/conf/acl/acl2014-1.html#TangWYZLQ14

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 947–953.

G Udny Yule et al. 1925. II.—A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FR S. *Phil. Trans. R. Soc. Lond. B* 213, 402-410 (1925), 21–87.

Cong Zhang and Jiangchuan Liu. 2015. On crowdsourced interactive live streaming: a Twitch.tv-based measurement study. In *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 55–60.

Ye Zhang and Byron Wallace. 2017. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 253–263.

11 SUPPLEMENTARY MATERIALS

11.1 Top 100 Emotes with Sentiment Votes

Table 15 shows all results from the emote sentiment survey, sorted by *Unknown/NA* answers in increasing order.

Table 15. Complete results of the Twitch Emote sentiment survey

Emote	Negative	Neutral	Positive	Unknown/NA
FeelsBadMan	71	17	19	1
FeelsGoodMan	1	7	98	2
LUL	11	23	72	2
OMEGALUL	17	26	62	3
PogChamp	1	3	101	3
:D	5	4	96	3
monkaS	27	59	18	4
POGGERS	4	13	87	4
Kappa	7	53	44	4
<3	1	4	99	4
Kreygasm	5	10	89	4
:)	6	14	84	4
HeyGuys	7	23	74	4
PepeHands	68	17	17	6
Pog	5	21	75	7
RIP	60	32	9	7
LULW	16	22	62	8
EZ	8	26	66	8
WutFace	64	17	19	8
BibleThump	59	19	22	8
Clap	5	23	71	9
PJSalt	68	13	18	9
SeemsGood	3	14	81	10
NotLikeThis	64	18	16	10
:p	7	57	33	11
SwiftRage	46	31	20	11
FailFish	72	10	15	11
HYPERBRUH	50	26	20	12
:(	70	15	11	12
TakeNRG	3	20	72	13
EleGiggle	11	32	51	14
FrankerZ	4	46	43	15
BabyRage	57	20	16	15
cmonBruh	57	21	15	15
BlessRNG	2	21	70	15
DansGame	58	20	15	15
AYAYA	9	25	58	16
Keepo	7	41	44	16
KappaPride	6	34	52	16
ResidentSleeper	55	20	17	16
;)	7	26	58	17
:o	13	59	18	18
Jebaited	25	27	37	19
KappaHD	8	33	48	19
VoteYea	4	22	61	21
D:	52	24	11	21
PowerUpR	2	40	44	22
4Head	17	18	51	22
PowerUpL	4	33	48	23
GivePLZ	3	23	59	23
VoteNay	46	28	11	23
CoolStoryBob	18	34	32	24
haHAA	45	19	20	24
KAPOW	6	42	35	25
TriHard	29	30	24	25
CoolCat	4	16	63	25
ANELE	40	34	8	26
KKona	21	40	21	26
MrDestructoid	13	52	17	26
:P	8	39	35	26
VoHiYo	8	22	52	26
TheIlluminati	13	52	16	27
MingLee	15	32	34	27
TwitchLit	7	34	39	28
HYPERS	7	18	54	29
BrokeBack	36	27	16	29
BloodTrail	7	29	42	30
OhMyDog	7	27	44	30
SMOrc	21	36	20	31
gachiBASS	12	23	41	32
cheer100	10	19	47	32
DoritosChip	16	42	17	33
OpieOP	18	29	28	33
:O	12	51	12	33
KonCha	9	23	42	34
PurpleStar	8	49	16	35
PunOko	37	32	4	35
PepePls	17	18	38	35
TwitchUnity	6	17	50	35
moon2A	43	22	5	38
CurseLit	8	36	26	38
SourPls	15	16	39	38
bleedPurple	11	33	24	40
SeriousSloth	11	34	22	41
Squid4	7	45	15	41
Squid2	6	44	16	42
sumSmash	22	31	12	43
lfsH	7	31	26	44
GachiPls	14	27	20	47
ninjaPon	8	30	21	49
FortOne	13	23	23	49
NaM	9	40	10	49
MercyWing1	8	33	18	49
GOWKratos	8	36	13	51
MercyWing2	7	34	16	51
mcaT	10	34	12	52
forsenPls	13	26	17	52
PepoDance	9	26	20	53
RedCoat	5	46	4	53
jinnytHype	7	31	17	53

11.2 Proof for Distribution Based Lexicon Approach from Section 5.1.2

With the previous notation from Section 5.1.2, that is $T^* = (t_1, \ldots, t_n)$ denote a tokenized and filtered Twitch comment, $C = \{\text{negative}, \text{neutral}, \text{positive}\}$ is the set of sentiment classes and under conditional independence $p(t_1, ..., t_n|c) = \prod_{i=1} p(t_i|c)$ it holds that

$$\operatorname*{argmax}_{c \in C} p(c|t_1, ..., t_n) = \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} p(c|t_i). \tag{1}$$

Proof.

$$\begin{aligned}
\operatorname*{argmax}_{c \in C} p(c|t_1, ..., t_n) &\overset{\text{Bayes}}{=} \operatorname*{argmax}_{c \in C} \frac{p(t_1, ..., t_n|c)p(c)}{p(t_1, ..., t_n)} \\
&\overset{\substack{\text{Irrelevant} \\ \text{constant}}}{=} \operatorname*{argmax}_{c \in C} p(t_1, ..., t_n|c)p(c) \\
&\overset{\text{Indep.}}{=} \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} p(t_i|c)p(c) \\
&\overset{\text{Bayes}}{=} \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} \frac{p(c|t_i)p(t_i)}{p(c)} p(c) \\
&= \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} p(c|t_i)p(t_i) \\
&\overset{\substack{\text{Factor} \\ \text{out}}}{=} \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} p(c|t_i) \prod_{i=1}^{n} p(t_i) \\
&\overset{\substack{\text{Irrelevant} \\ \text{constant}}}{=} \operatorname*{argmax}_{c \in C} \prod_{i=1}^{n} p(c|t_i)
\end{aligned}$$

□