E^{2}T^2: *E*mote *E*mbedding for *T*witch *T*oxicity Detection

Korosh Moosavi
University of Washington
Bothell, Washington, USA
moosavik@uw.edu

Elias Martin
University of Washington
Bothell, Washington, USA
eamart34@uw.edu

Muhammad Aurangzeb Ahmad
University of Washington
Bothell, Washington, USA
maahmad@uw.edu

Afra Mashhadi
University of Washington
Bothell, Washington, USA
mashhadi@uw.edu

ABSTRACT

The Internet has become the medium of choice for socialization and communication. The rise of live streaming services has created countless online communities of varying sizes with their own jokes, references, slang, and other means of communication. One of the largest live streaming services is Twitch.tv or Twitch, where a unique culture of niche language and emote usage has developed. Emotes are custom-made images, or GIFs, used in chat with varying degrees of access influenced by channel and external site subscription status. Emotes render standard forms of English Natural Language Processing (NLP) for tasks such as detection of toxicity or cyberbullying ineffective on Twitch. In this paper, we propose a methodology and offer a largely-trained dataset for detecting emote-based toxicity on live streaming platforms such as Twitch.

CCS CONCEPTS

• **Information systems** → *Social networks*; • **Human-centered computing** → **Collaborative and social computing design and evaluation methods**; • **Security and privacy** → Social aspects of security and privacy.

KEYWORDS

Toxicity Detection; Twitch; Natural Language Processing; Collaborative and Social Network

ACM Reference Format:
Korosh Moosavi, Elias Martin, Muhammad Aurangzeb Ahmad, and Afra Mashhadi. 2024. E^{2}T^2: *E*mote *E*mbedding for *T*witch *T*oxicity Detection. In *Companion of the 2024 Computer-Supported Cooperative Work and Social Computing (CSCW Companion '24), November 9–13, 2024, San Jose, Costa Rica.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3678884.3681840

1 INTRODUCTION

Twitch is the leading live streaming platform, surpassing competitors like YouTube and Kick. Twitch has even gained mainstream attention from politicians and large corporations. Twitch experienced significant growth in recent years which doubled its average concurrent viewers and watch-hours [9]. Twitch's unique culture of communication is centered around custom emojis called *emotes*. Initially, a few global emotes replaced common 'faces' such as :) and :(in the chat.

Currently, there are over 200 available free emotes and many more unique emotes per channel. Subscribing to channels unlocks these custom emotes and allows the subscriber to use the emote on any channel in the same way as a global emote. This history and low barrier to customization has led to inside jokes, trends, recognizable celebrity faces, and many emotes that overlap in visuals and meaning with varying levels of nuance. This is made more complex by the popularization of 3rd-party extensions such as BetterTTV (BTTV) and FrankerFaceZ (FFZ) which introduce their own libraries and assortments of emotes for streamers to choose from. However, the emote culture has also led to some instances of emotes being misused to promote cyberbullying and toxicity. For example, the TriHard emote has been used in racist contexts or certain imagery such as the Confederate flag have brought attention to the toxicity in Twitch chats.

The largest prior study on live stream data is [11], which trained a Word2Vec model on Twitch chat data to analyze the sentiment of the 100 most popular emotes across the platform. However, their dataset is limited to only the top 100 global emotes only. They demonstrated through an extensive analysis of a massive Twitch chat corpus that this domain is unique enough in its language pattern that Natural Language Processing (NLP) methods used for other online domains, even other short-response informal domains like Twitter, are insufficient for capturing the information and context in Twitch's live stream chats. They showed that emotes carry a large amount of information which shapes the meaning of the message in its entirety. By comparing the performance of different lexica (VADER for social media texts [8], Emoji for traditional standardized emoji [14], and Emote for their labeled emotes) Kobs et al. show that not only are emotes highly prevalent in the corpus, but that their influence on the sentiment of the message is greater than either of the other two lexica. However, there is currently no work that has examined relying on the embedding space to discover toxicity across channels by leveraging the associations between emotes.

In this paper we propose a new embedding space, constructed based on the collection of both private and global emotes from 2379 channels on the Twitch platform. We describe the steps taken to collect, process, and create vector embeddings of these emotes to

CSCW Companion '24, November 9–13, 2024, San Jose, Costa Rica.

ACM ISBN 979-8-4007-1114-5/24/11
https://doi.org/10.1145/3678884.3681840

construct an extensive embedding space of emotes. Furthermore, we propose a process for detecting toxicity by utilizing this emote embedding space. Our resulting dataset[1] includes raw data, embeddings, and labeled information regarding the tokens seen in chat logs. We then construct a pipeline for detecting toxic emotes by employing a state-of-the-art toxicity detection NLP-based model to extract a subset of emotes that are frequently accompanied by toxic English text. We use these seed emotes to discover other toxic emotes in new Twitch domains through embedding association that may have gone undetected otherwise.

2 METHODS

2.1 Data Collection for Embedding Space

Prior to this study, only a small handful of Twitch chat datasets were made publicly available (Table 1). These include [17], a study demonstrating the capability of Word2Vec models to learn from Twitch chat data; [10], a dataset uploaded to Harvard Dataverse; and [11], a large study performing sentiment analysis on Twitch emotes. As described in Table 1, these studies all pre-date the global pandemic, during which Twitch had a surge in online activity. Given the age of the previous datasets and the factors affecting the rate that emote usage and meaning can change, prior datasets were not included in the training corpus. Instead, the embedding space of this project is compared to the open-source embeddings provided by Emote-Controlled [11] to shed light on how the language of emotes differs now from almost five years prior.

Data for the embedding space portion of this project was collected during the first week of December 2022. We chose to scrape a small sample from the most popular channels on the platform, which would be representative of the largest chat communities. The top 2500 English-speaking channels were selected, sorted by watch time (hours), as reported by [5]. A Python script was used with the TCD (Twitch Chat Downloader) package [13] on each channel, downloading the 10 most recently saved broadcasts for each channel. In the end, this resulted in 2379 channels of data, as some channels were banned, changed names, or otherwise had become unavailable. Of the remaining channels, not all had 10 broadcasts, as some channels chose to selectively remove their broadcasts from their archive or chose to not have their broadcasts archived at all. Because of this, the total archived broadcasts were 22,977 livestreams, resulting in approximately 8GB of raw chat data in text files. For the same reason, the archived streams that are being analyzed have some variance in recency between channels (i.e. some channels still have broadcasts from 2021 in their 10 most recently archived broadcasts).

2.2 Data Preprocessing for Embedding Space

For the purposes of training, the corpus was tokenized using the tokenizer published by Kobs et al. [11], as it incorporates extensive use of regular expression (Regex) patterns to filter text from the Twitch domain specifically. Additionally, using the same tokenizer allows for greater confidence in the results of the embeddings being an outcome of the data itself rather than a difference in data handling when comparing the two. The tokenizer recognizes ASCII emoticons and arrows, HTML tags, Twitter-like hashtags, standard grammar syntax, chat commands, and negation, as well as a lexicon of some of the most popular emotes at the time of the study.

Although there were many typos and conjugations for words, the unique vocabulary of the Twitch chat domain makes advanced preprocessing strategies highly unreliable and susceptible to mistakes. For this reason, additional steps such as stemming and lemmatization were not performed on the training corpus. However, the corpus vocabulary was stemmed in addition to the preprocessing for the purposes of data exploration and visualization to aid in understanding the type of content being discussed in these livestream chats.

2.3 Creating the Embedding Space

Word2Vec differs from most of the recent developments in large language models (LLMs). Models such as BERT, BART, GPT, and their derivatives use a Transformer architecture which includes Attention layers, introduced in 2017 [18], that allow these models to learn contextual information about words and their syntax to create dynamic embeddings. Given the recent popularization of LLM models, work is beginning to explore how they may be leveraged for recommendation tasks. Currently, research into these LLMs being used as recommenders largely revolves around prompt design [1, 15] and fine-tuning model weights [7]. While these approaches allow for more flexibility and detail in model responses, the results are not very explainable and may vary greatly, with early evidence of biases related to non-binary communities [12].

Word2Vec could be considered one of the earliest NLP models, introduced by Mikolov et al. in 2013 [16], and relies on static embeddings, meaning that every word is represented in a single way independently of the context in which they are used. Although this older method of language embedding struggles with more complex tasks such as conversational queries and multiple-word representations, it allows for easy and direct comparisons between words, or in our case, emotes.

Model training was performed using Gensim's implementation of Word2Vec with a CPU (i7-9700K @ 3.60GHz) on a Windows PC with 64GB of RAM. Memory limitations were overcome by separating tasks into batches. The model uses the default Skip Gram Negative Sampling (SGNS) architecture. As one of the end goals of creating this embedding space is to predict the emotes associated with a target emote, SGNS is the natural choice of architecture for this task, in contrast with the Continuous Bag of Words (CBOW) architecture which aims to predict a word given context words.

The final model used for assessments and visualizations was trained for 20 epochs with 500 vectors, a window size of 5, and a minimum occurrence rate of 10. The goal of these parameters is to capture as much information as possible on a much smaller corpus size. Chat message lengths are often around 5 words in length, so a window size of 5 allows for adequate context capture for this project. This resulted in a 4.1GB model with 733K keys, in contrast with the model provided by [11] which was trained with 128 vectors on 2.4M keys.

An important test of the embedding space is a test of the associations that the model has learned to make. The classic demonstration of this is to use the 'Man is to Woman as King is to __' example.

[1] 10.5281/zenodo.8012284

Datasets	Duration	Size	Channels
Emote-Controlled [11]	4/2018 - 6/2018	~300GB	global
Harvard Dataverse [10]	4/2018 - 6/2018	~12GB	52 channels
TwitchChat Dataset [17]	6/2019 - 10/2019	~1GB	666 channels
Emotes-2-Vec (our dataset)	**10/2022 - 11/2022**	**~8GB**	**2379 channels**

Table 1: Prior datasets on Twitch chat logs

For this example, both models provided comparable results, with 'queen' being the strongest recommendation. Emote-Controlled [11] also made a similar comparison of ':)' is to :(' as FeelsGoodMan is to __', for which both embeddings recommended 'FeelsBadMan', however, the 2022 embeddings additionally recommended some other 'Pepe the Frog' emotes which also signify sadness, such as 'PepeHands', 'PeepoSad' and 'sadge'.

3 TOXICITY DETECTION FRAMEWORK

In the past decade, toxicity and hate-speech detection has received a considerable amount of attention from the research community [2–4, 6, 19]. In particular, many works have focused on examining and training Transformer based models to learn toxicity in natural language. These models are trained on the Jigsaw dataset series[2] from 2018-2020 which is a collection of comments from Wikipedia where the labels annotated by humans correspond to toxic and non-toxic. As we will demonstrate next, a limitation of these models is that they require a well written, structured sentence in English and are sensitive to the length of the text. These models fall short to detect toxicity (i.e., high False Negative Rate) when applied to chat messages in Twitch as these messages are often short and contain the use of multiple emotes. Toxicity can go undetected especially in cases where emotes are stitched together to alter their appearance or meaning for some devious purpose, such as when used to convey an identity attack on an individual.

To this end, we aim to build a pipeline to detect toxic messages through associations learned from the emote embedding space. We first rely on a NLP classifier to detect a subset of chat messages that are toxic, which we then extract emotes from the toxic messages and label them as *seeds*, and through the association in the embedding space, we use the seeds to discover other potential toxic emotes in other channels. These steps are covered in detail subsequently.

3.1 NLP-based Toxicity Classifier

We used an state-of-the-art toxicity classifier available on HuggingFace[3]. This model is a RoBERTa based model that was trained on three Jigsaw datasets from 2018-2020 of over 2 million Wikipedia comments that were labeled for different subsets of toxicity by human annotators.

To discover the first set of toxic emotes in our dataset, henceforth referred to as *seeds*, we collected raw chat logs from TwitchAPI[4]. In selecting a Twitch stream suitable for this task, we selected a channel that is rated as one of the top 10 controversial channels in the Twitch community for covering controversial topics, which in turn sparks controversial and sometimes toxic comments in the stream chat logs. One such streamer is Hasan Abi[5], whose entire stream chat logs were recorded over a 10 day period from November 17th - November 26th, 2022. From this resulting data set, an arbitrary day was selected to serve as the training stream data.

Since the majority of chat messages on Twitch consist of relatively few words, we filtered stream comments that had less than eight words in them. This was in an effort to retain comments with a higher context that would enable more accurate classification by the NLP model. This resulted in a relatively robust data stream of about 11,500 chat comments. Moreover, data was also collected on the different types of emotes that occured in the stream, including global, 3rd-party extension, and Hasan Abi's channel-specific emotes. In these 11.5k chat comments, there were 7782 total instances of emote usage corresponding to 299 unique emotes. The majority of these emote usage instances ($N = 6382$) were related to 210 unique channel-specific emotes, that can only be unlocked by subscribing to the Hasan Abi channel, as well 89 unique global emotes.

Out of these 11.5K stream chat comments, we found 14% of those messages to be toxic ($N = 1668$) as classified by the Roberta toxicity classifier model.

3.2 Human Validation

To validate the recall of the classifier, we took a random sample of 500 messages from non-toxic chat messages and manually inspected them to see whether they contained toxic messages that may have been undetected. We found that 14% of those messages were indeed **toxic**. This confirms our initial hypothesis of the difficulty in classifying Twitch messages as toxic and indicates the inherent limitation of applying NLP-based toxicity detection models to Twitch chat messages. We overcome this limitation through our proposed pipeline. Table 2 shows examples of False Negative chat messages, demonstrating the subtlety and difficulty of cases that were detected by a human annotator as toxic but were missed by the model.

3.3 Seed Detection Using Toxicity Ratio

In addition to False Negative cases, as Table 2 illustrates there are cases that the classifier detects *neutral* messages as toxic (i.e., False Positive). These cases often occurred due to the model being overly sensitive to messages containing profanity, or addressing controversial topics such as mental health or death discussed amongst Twitch users. To account for these errors, we extracted emotes from all the chat messages, so to measure which emotes are used predominantly in a toxic way and with what associated frequency.

[2]Jigsaw Toxicity Datasets: Jigsaw 2018, Jigsaw 2019, Jigsaw 2020.
[3]https://huggingface.co/s-nlp/roberta_toxicity_classifier
[4]https://dev.twitch.tv/docs/api/
[5]https://www.twitch.tv/hasanabi

Text	Human	Classifier	Error
Faded than a ho	Toxic	Neutral	FN
Lmao tell QT she has Nazi merch I guess	Toxic	Neutral	FN
we are all mentally ill this is twitch	Neutral	Toxic	FP
did you die and turn into a ghost?	Neutral	Toxic	FP

Table 2: Twitch Comment Classification Examples

To this end, we calculate "*toxicity ratio*" as the number of times an emote has been detected in a toxic message divided by total number of times it has been used across the chat corpus (i.e., all the messages). A toxicity ratio of 1 would indicate that an emote has been seen exclusively in toxic messages, whereas a ratio of 0 indicates that the emote has never been used in a toxic message.

Figure 1 presents the toxicity ratio score versus the frequency. We find that some of the Hasan Abi channel specific emotes such as hassammie ( ) and punchies ( ) are associated mostly with toxic messages, whereas gigahas ( ) is also a channel emote but is used more evenly in toxic and non-toxic messages. Conversely, Sussy ( ) is a 3rd-party extension emote that is accessible globally outside of the Hasan Abi channel, but appears frequently in toxic ways inside of the channel.

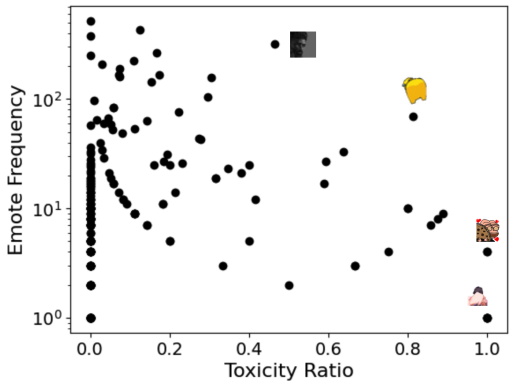

Figure 1: Distribution of Detected Emotes

3.4 Channel vs Global Emotes

Figure 2 presents the histogram distribution of all emotes, global emotes and channel specific emotes. Upon inspection, we see that the resulting distributions of emotes across global emotes and Hasan Abi channel's specific emotes share similar attributes and overall shapes (Jensen-Shannon distance 0.260, 0.339, 0.102 between global and all, global and channel, and channel and all emotes respectively). However, it is considerable that the channel specific emotes explored a greater range of the toxicity spectrum and clustered more towards toxicity based on their usage in chat as opposed to how the globally available emotes were used. These channel specific emotes were used considerably more often with toxic connotations with chat comments than were the global emotes. This exemplifies that channel specific emotes can deviate from the traditional usage patterns of globally available emotes, and proves that exploring these domain specific emote languages to detect toxicity can be both promising and difficult due to the additional nuance in the way emotes are used between channels. Several approaches will be suggested to explore this toxicity detection in new domains in the subsequent section. Moreover, the distribution of toxic channel emotes illustrates that the Hasan Abi channel was a suitable candidate for this toxicity detection as this channel's emotes display a notable difference in toxic usage than the global emotes.

3.5 Toxic Emote Discovery Through Association

Using the embedding association described earlier, we can find similar emotes to any given channel-specific emote. To this end, our embedding space can be used to find similar emotes to the detected channel toxic seeds from the previous step. For example, if we look up the association of the Hasan Abi emote *hassammie* ( ), which we found to be used highly in toxic conversations, we can detect similar channel specific emotes in the Pokimane channel such as poki emotes 1-4[6]. Whether these emotes in Pokimane channel are actually also used in toxic conversations is a research question that we aim to study in future.

Indeed, discovering toxic emotes through the embedding space could enable the research community to explore the following: 1) learn the combination of emotes that are used to convey toxicity or cyber-bullying; 2) study whether toxicity is transitive across channels and communities. We will describe these research directions next.

4 DISCUSSION

In this work, we created and have made available a new embedding dataset of Twitch emotes by following reproducibility practices and is permanently hosted on the following DOI (10.5281/zenodo.8012284). We proposed a methodology based on NLP classifiers and an embedding space to discover toxic emotes on Twitch by minimizing false negative and false positive cases, and demonstrated how these toxic seeds can be used to discover other toxic emotes across different channels. Our code repository is also open-sourced. [7]

4.1 Implication

We believe our work has the following implications for the research community and practitioners.

[6]https://twitchemotes.com/channels/44445592
[7]https://github.com/KoroshM/Emote-Recommender.

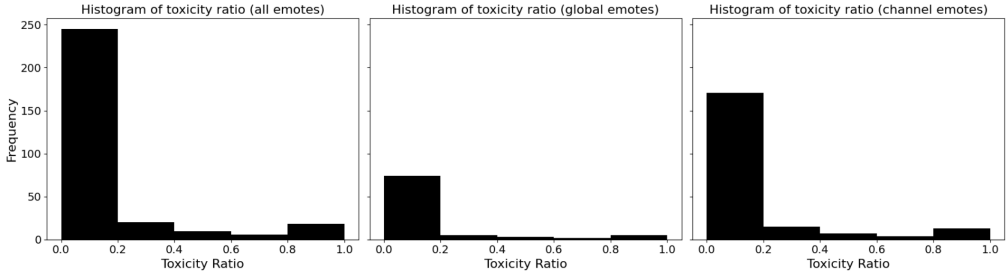

Figure 2: Histogram distribution of toxicity of a) all emotes, b) global emotes, and c) channel-specific emotes

4.2 Theoretical Implication: Toxicity Pattern Across Communities

The proposed methodology and the adjunct embedding space dataset can facilitate toxicity detection in new live streaming communities, enabling the research community to explore user behaviour across communities. Our methodology can be used to address questions such as whether the detected toxic global emotes are used consistently in a toxic way across different communities. For instance, those with different streaming topics or even across different languages. In addition to the language usage, we believe that our research has an implication on understanding how users engage in cyberbullying. Our embedding space and discovered toxic seeds can be used not only to label toxic messages and moderate conversation, but also study the behaviour of those users who engage in toxic conversations across channels and communities.

4.3 Practical Implication: An Active Learning Moderating System

A practical implication of our work is that it can be integrated into a live moderation system that could learn the associations of newly discovered toxic seeds spontaneously and assign a toxicity score to every emote in the embedding space. As our results have shown, channel-specific emote messages are more likely to be used in toxic conversations. We believe this is an important implication of our work that can better accommodate for language evolution on Twitch. We believe such a system in the future could be augmented with LLM applications to provide more context as to why some chat messages were labelled as toxic by NLP techniques, and utilize a human-in-the-loop approach to increase the transparency behind toxicity classification decisions. Thus, creating an artificial intelligence and human moderation system.

4.4 Limitations and Future Work

While the described toxicity detection feature could prove invaluable across live streaming communities such as Twitch, it also necessitates human validation to ensure that it does not become skewed by either initial or exploratory biases.

Accordingly, after the initial population of the toxic seed emotes, a human annotator who is a domain expert in the initial channel would be required to ensure that these initial emote scores are relatively accurate, and do not cause a chain reaction of misinformed biased propagation.

Finally another limitation of this study is the state-of-the-art NLP model employed had noticeable biases and could be overly sensitive to comments including references to topics including mental health, death, religion, and others even if they were not necessarily used in an obvious toxic connotation. For example, the NLP model was sensitive to comments mentioning death like "did you die and turn into a ghost? 👻". Or similarly, it was sensitive to mental health which is visible through examples like "WHAAAT YOUTUBE CHAT IS CRAZY." Messages containing religious connotations like "THESE GOD-DANG COFFEE MAKERS THINK THEIR JOB IS HARD" were also likely flagged for mentioning the word "god". Evidently, these messages do not seem inherently toxic in meaning, but it seems plausible that the NLP model could be overly cautious when classifying known sensitive topics such as those mentioned previously. We anticipate that by implementing an active learning moderating system as discussed previously, this could serve to mitigate some of this sensitive topic bias by promoting discussion as to why something was classified as toxic and increasing the transparency behind decision making.

5 CONCLUSION

In this paper, we created a new multi-channel embedding space of Twitch emotes, and showed an example use case of how a toxicity detection pipeline can be constructed to leverage both the best of NLP classification methods and the embedding space to discover other toxic emotes across channels. We have also shown through our analysis the unique usage of toxic language that exists within individual Twitch channels, and how this community toxicity adds an additional level of nuance to detecting toxicity within these micro communities. Furthermore, we put forward a novel methodology to explore and detect these instances of toxicity within individual Twitch channels, which we hope proves valuable to the research community exploring toxicity detection on live streaming platforms such as Twitch. The embedding dataset and our methodology can be used by the research community to examine associations in the embedding space, and study toxicity across communities.

REFERENCES

[1] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. https://doi.org/10.48550/arXiv.2305.00447

[2] Nicole A Beres, Julian Frommel, Elizabeth Reid, Regan L Mandryk, and Madison Klarkowski. 2021. Don't You Know That You're Toxic: Normalization of Toxicity in Online Gaming. In *Proceedings of the 2021 CHI Conference on Human Factors in*

Computing Systems (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 438, 15 pages. https://doi.org/10.1145/3411764.3445157

[3] Jie Cai, Sagnik Chowdhury, Hongyang Zhou, and Donghee Yvette Wohn. 2023. Hate Raids on Twitch: Understanding Real-Time Human-Bot Coordinated Attacks in Live Streaming Communities. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (2023), 1–28.

[4] Jie Cai, Cameron Guanlao, and Donghee Yvette Wohn. 2021. Understanding rules in live streaming micro communities on twitch. In *Proceedings of the 2021 ACM International Conference on Interactive Media Experiences*. 290–295.

[5] David. 2022. Most Watched Twitch Channels - Stats and Analytics. Past 90 Days. https://sullygnome.com/channels/90/watched.

[6] Lukas Dreier and Johanna Pirker. 2023. Toxicity in Twitch Live Stream Chats: Towards Understanding the Impact of Gender, Size of Community and Game Genre. In *2023 IEEE Conference on Games (CoG)*. 1–4. https://doi.org/10.1109/CoG57401.2023.10333159

[7] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. https://doi.org/10.48550/arXiv.2303.14524

[8] C. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media* 8, 1 (May 2014), 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

[9] Mansoor Iqbal. 2023. Twitch Revenue and Usage Statistics. https://www.businessofapps.com/data/twitch-statistics/.

[10] Jeongmin Kim. 2019. Twitch.tv Chat Log Data. https://doi.org/10.7910/DVN/VE0IVQ

[11] Konstantin Kobs, Albin Zehe, Armin Bernstetter, Julian Chibane, Jan Pfister, Julian Tritscher, and Andreas Hotho. 2020. Emote-controlled: obtaining implicit viewer feedback through emote-based sentiment analysis on comments of popular twitch. tv channels. *ACM transactions on social computing* 3, 2 (2020), 1–34. https://doi.org/10.1145/3365523

[12] Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in Large Language Models. In *Proceedings of The ACM Collective Intelligence Conference (CI '23)*. Association for Computing Machinery, New York, NY, USA, 12–24. https://doi.org/10.1145/3582269.3615599

[13] Petter Kraabol. 2022. Twitch Chat Downloader. https://github.com/PetterKraabol/Twitch-Chat-Downloader.

[14] Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of Emojis. *PLOS ONE* 10, 12 (12 2015), 1–22. https://doi.org/10.1371/journal.pone.0144296

[15] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. A First Look at LLM-Powered Generative News Recommendation. https://doi.org/10.48550/arXiv.2305.06566

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR* (2013). https://doi.org/10.48550/arXiv.1301.3781

[17] Charles Ringer, Mihalis Nicolaou, and James Walker. 2020. Twitchchat: A dataset for exploring livestream chat. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 16. 259–265.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. https://doi.org/10.48550/arXiv.1706.03762

[19] Yingfan Zhou and Rosta Farzan. 2021. Designing to stop live streaming cyberbullying: a case study of twitch live streaming platform. In *Proceedings of the 10th International Conference on Communities & Technologies-Wicked Problems in the Age of Tech*. 138–150.