

Towards Predicting the Subscription Status of Twitch.tv Users

ECML-PKDD ChAT Discovery Challenge 2020

Konstantin Kobs,¹ Martin Potthast,² Matti Wiegmann,³
Albin Zehe,¹ Benno Stein,³ Andreas Hotho¹

¹Julius-Maximilians Universität Würzburg
{kobs, zehe, hotho}@informatik.uni-wuerzburg.de

²Leipzig University
martin.potthast@uni-leipzig.de

³Bauhaus-Universität Weimar
{matti.wiegmann, benno.stein}@uni-weimar.de

<https://events.professor-x.de/dc-ecmlpkdd-2020/>

Abstract We investigate whether the subscription status of active users of Twitch can be inferred from their activity patterns in the chats of streamers. To enable a diversity of solutions to this problem, this task was advertised as an ECML-PKDD discovery challenge 2020, called Chat Analytics for Twitch (ChAT). Four participants submitted their working prediction models, which were evaluated at our site. The winning approach achieved an F_1 score of 0.343, outperforming the baseline by a significant margin. The most salient conclusion that can be drawn at this time is that interaction behavior plays a crucial role in solving this task, meriting further analysis into this direction.

1 Introduction

The popularity of game streaming and corresponding platforms, such as Twitch,² for entertainment and professional e-sports is on the rise. The basic function of such platforms is to enable a screencast along with commentary to a live audience. While most streamers focus on streaming and commenting the gameplay of games they are currently playing, other video and audio content is also streamed, though at a much smaller volume. The audience, in turn, can interact with the streamers in a channel's chat and by other means. This enables streamers to engage with their audience in real time in order to build a followership and, eventually, to monetize their channel. The audience can donate to streamers and subscribe to the channel by paying a monthly fee, which is typically between

Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

² <https://www.twitch.tv>

five to twenty-five dollars per channel subscription at the time of writing. This comes with exclusive chat and channel features such as exclusive channel-specific emotes, which are still or moving images approximately the size of a standard emoticon. A channel’s earnings are split 50/50 between the streamer and Twitch, incentivizing both to convert watching users to subscribed users [12]. Targeted advertising can be a useful tool to accomplish this, but depends on identifying users open to subscribe to a channel. If a classifier can be developed that predicts the subscription status of a user-channel combination (based on the chat comments and activity patterns of the user in the channel), then applying this model to currently unsubscribed user-channel combinations can result in potential targets for advertisement. At the ECML-PKDD Discovery Challenge “Chat Analytics for Twitch” (ChAT), the task was to build such a binary classification system.

In order to enable the task, we provide a large training dataset consisting of over 400 million public Twitch comments published along this novel task. Additionally, we constructed a test dataset with certain characteristics that can be used as both a benchmark for comparison and as a basis for future research and analysis. The training dataset was provided to participants, who developed their prediction models at their own site and then deployed their trained model to our online evaluation platform TIRA [19]. Each team was assigned a virtual machine to facilitate the installation of required dependencies, where at no point participants were given direct access to the test data. TIRA enables blind evaluation by preventing outside access while a software is executed. A participant’s software was executed remotely on the test data and its outputs were recorded and evaluated. The virtual machines on which the participants deployed their software were archived for reproducibility purposes. After the submission deadline all data has been made publicly available.

This paper describes the task and the evaluation data for this challenge, summarizes the submissions, and presents an analysis of their performance. Our contributions are threefold: (1) An original task to predict the subscription status of Twitch users at given channels based on their interactions and chat messages. (2) A training dataset with Twitch chat messages, and a test dataset with characteristics useful for further analysis. (3) An overview and evaluation of the approaches that were submitted as part of the challenge. The remainder of the paper is organized as follows: Section 2 gives an overview on related work. Section 3 introduces the task and the datasets as well as the evaluation metric and the baseline. Section 4 describes the approaches developed by the participants, including the employed features and models. Section 5 analyzes the results of the submitted approaches.

2 Related Work

As Twitch is one of the most popular streaming platforms for games, numerous studies have been conducted to understand the use, the impact, and the challenges of this new form of media. While many publications examine social,

cultural, and economic dynamics of the platform [1, 5, 8, 14, 24], implications for the media and community landscape [7, 12, 13, 22, 23], and its language [15, 18], only few also utilize machine learning methods to automatically process the chat and interaction data [2, 15].

Kobs et al. investigate the usefulness of emotes as indicators for the sentiment of chat messages [15]. They build an emote sentiment lexicon via crowdsourcing to improve sentiment analysis models based on dictionaries and convolutional neural networks, showing that common word-based dictionaries cannot capture the sentiment in the setting of chat messages on Twitch due to the platform-specific slang. Barbieri et al. define two Twitch-specific tasks: predicting emotes that are likely to be used in a message, and detecting troll messages, for which they also utilize emotes to generate ground truth labels [2]. Both tasks focus on the text of chat messages. Multiple experiments show that a Bidirectional LSTM [11] performs best for both tasks.

In this work we introduce a novel task: the prediction of the subscription status of users based on their channel interaction; obviously this is valuable knowledge that can be used in a subscription recommendation setting. Many recommendation algorithms, such as collaborative filtering, try to predict whether a user is interested in an item (in our case a channel) by correlating her preferences with similar users [21]. In our setting, users who have subscribed to similar channels might be recommended the channels to which other similar users are subscribed to. However, personal interactions are not trivial to be included in such algorithms. For example, Twitter relies on a graph-based recommendation system that models users another user follows [10]. YouTube recommends individual videos instead of entire channels. Twitch subscriptions cost a monthly fee, while other social media platforms such as Facebook or Instagram allow users to follow or subscribe to channels free of charge. Note that approaches outside of the domain of social media are also related: E.g., in a digital newspaper setting, potential subscribers are identified using different user engagement features, such as the number of articles read, or the average time spent on an article [6]. Instead of a single newspaper, Twitch hosts hundreds of channels that a user can subscribe to. The direct interaction of users with their channels' hosts based on the chats and comments hence play an important role.

3 Subscription Prediction for Twitch


We define the task of subscription prediction for Twitch as follows: Given the chat messages of a user in a channel on Twitch (including metadata such as timestamps and the currently streamed game), predict whether or not the user is subscribed to the channel. We instructed the participants not to augment their models using data other than the training data we supplied. The only exception to this rule is the use of pretrained models or dictionaries of emotes and their text representations. We ensured that the training and test datasets are disjoint

in terms of user-channel combinations in order to prevent leakage of ground truth.³

3.1 Twitch Crawl

A large dataset of nearly all publicly available Twitch comments in January 2020 was crawled using Twitch’s official API. Only channels labeled as English were considered. All user-channel combinations for which the subscription status changed during the recorded time period were omitted, i.e., if a user subscribed to or unsubscribed from the channel during January 2020. For each user-channel combination for which at least one comment has been recorded, the following metadata was recorded:

- Name of the channel (anonymized).
- Name of the commenting user (anonymized).
- Whether or not the user is subscribed to the channel.
- All public chat messages of the user in the channel in the recorded time period, each containing the timestamp when the user commented, the game that was played in the channel when the user commented in the form of a string label, and the chat comment/message itself.

Many messages contain so-called emotes, i.e., still or moving images approximately the size of a standard emoticon. Emotes are very popular on Twitch and are used to express the emotional state of a user while watching [15]. Every emote has a text representation that is present in the text field. For example, in the message “awesome LUL”, ‘LUL’ is the text representation of the emote , which indicates general laughter and amusement.

The crawled dataset was split into training and test datasets. The training dataset has a size of approximately 37 GB. In what follows, we describe the sampling strategy for the test dataset and report on a brief corpus analysis.

3.2 Test Dataset

For the test dataset, we sampled 90,000 user-channel combinations with corresponding metadata, resulting in 636,452 messages. To ensure that different user and channel activities are covered in the test dataset, we employed the following sampling procedure. Given the number of messages in the dataset per user and channel, we categorized each user and channel into three activity classes based on the number of comments: 25 % of the users/channels with the lowest message counts are considered to be of low activity, and 25 % with the largest message counts are considered to be of high activity. All other users and channels are considered to be of normal activity. We sampled 10,000 user-channel combinations

³ By mistake, emotes accessible to already subscribed users were not removed; one of the participants exploited this “feature” (without notifying us), rendering their approach infeasible in practice. Nevertheless, it provides for an interesting baseline.

Table 1. Statistics of the training dataset.

Statistic	Mean	Standard Deviation	Median
channels a user comments in	3.73	14.91	2
channels a user is subscribed to	1.50	1.33	1
comments per user in channel	51.83	2350.20	7
comments per user in channel (subscribed)	55.66	164.78	9
comments per user in channel (not subscribed)	43.08	2385.56	6
comments in a channel	2802.61	23931.08	285

for each combination of user and channel activity, yielding 90,000 user-channel combinations in total for the test dataset.

These user-channel combinations were removed from the training dataset. Additionally, for a randomly sampled 50 % of users in the test set, we removed their comments in other channels as well, such that half of the users are not present in the training dataset at all. This allowed us to analyze whether models perform better for already known users, even though no messages in the desired channel were present.

3.3 Training Dataset

After removing the test data and the messages of 50 % of the test users in all other channels, the training dataset contained 29,539,420 user-channel combinations and a total of 410,686,442 public Twitch comments. Table 1 overviews key figures of the data. The training dataset has records of 146,537 channels and 7,923,774 users. On average, each user has written comments in fewer than four channels, while being subscribed to an average 1.5 channels. Among the 29 million user-channel combinations in the training data, 2,368,323 (8.02 %) users were subscribed, and 27,171,097 (91.98 %) were not. Subscribed users have a higher mean comment count than non-subscribed users. The high difference of standard deviation between subscribed and not subscribed users can be explained by the use of bots that are not subscribed to channels, but comment very often in order to engage viewers, or to notify users and the streamer about certain events.

Figure 1 depicts the number of messages that a user has sent or a channel has received. Both histograms show exceedingly active users and channels having sent and received an extensive number of comments. The user with the most comments, “streamelements”, is a bot used by many streamers to send notifications about the channel to the chat. The channel that received the most comments, “xqcow”, belongs to a professional gamer and streamer.

Regarding the content, the word cloud in Figure 2 gives an impression of the words used in the comments. Emotes play an important role in Twitch comments, e.g., 🤪 LUL and 🧑 PogChamp are heavily used. The Twitch chat is case-sensitive and only displays emotes if they are typed correctly; emotes mostly contain capital letters. Therefore, most words in Figure 2 containing capital

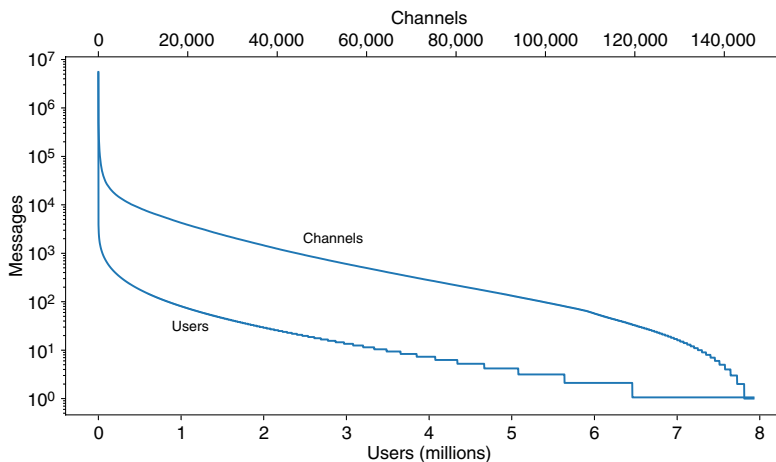


Figure 1. Histogram of the number of messages per channel (top) and user (bottom).

letters depict emotes; “normal” words are mostly written lower-case. Besides emotes, ASCII-style emoticons such as :) or :D are popular on Twitch, too. It is not surprising that gaming and online slang words, such as “u” as a short form for “you”, “stream”, “play”, and “lol”, are often used. Kobs et al. [15] provides a more detailed analysis of the Twitch comment’s usage and activity patterns.

3.4 Performance Measures and Baselines




Given the 90,000 user-channel combinations including their metadata from the test dataset, the participants’ models were supposed to predict whether or not the user is subscribed to the channel. Submissions were evaluated using the F_1 measure, which is the harmonic mean of precision and recall with respect to subscribed user-channel combinations. Owing to the high class-imbalance between subscribed and unsubscribed users, a majority baseline yields an F_1 score of 0.0. We further provide a random baseline which assigns class labels according to their distribution found in the the training dataset (8.02 % subscribed, 91.98 % not subscribed). Finally, the submission 🚩 ItsBoshyTime provides a baseline based on the usage of subscriber-only emotes.



4 Survey of Submitted Approaches

From the 23 registered teams only four submitted their approaches, three of which submitted also a notebook paper describing their approach.⁴ All approaches rely on certain machine learning methods (see Table 2 for an overview),


⁴ Given the importance of emotes on Twitch, and for a bit of fun, we asked participants to choose one of the most common Twitch emotes as their team name.




Table 3. Overview of the hand-crafted features for each approach, separated by stylometric features of a users messages, activity features of the user and the channel separately, and interaction features of a user within a channel.


Features	 VoyTECH	 CoolStoryBob	 StinkyCheese
Stylometric (user)	med. chars/message	num. emoji	avg. num. words/channel
	max. chars/message	num. distinct emotes	–
	num. chars/message	num. emotes	–
	–	num. allcaps	–
	–	num. single chars	–
	–	num. numericals	–
	–	num. stop word	–
	–	num. !, #, @	–
	–	num. emoticons	–
	–	avg. word length	–
–	sentiment score	–	–
Activity (user)	activity group	num. word	num. words
	num. channels	num. distinct games	num. channels
	num. messages/game	num. games	–
	max. message interval	–	–
	active days/month	–	–
	num messages/game	–	–
	num. games	–	–
	top game	–	–
	num. message/chatting	–	–
	% messages in top game	–	–
Activity (channel)	id	–	num. messages
	num. users	–	avg. num. words per user
	activity group	–	agv. users
	max. message interval	–	–
	active days/month	–	–
	num messages/game	–	–
	num. games	–	–
	top game	–	–
	num. message/chatting	–	–
	% messages in top game	–	–
Interaction (user + channel)	time of first message	–	sum. time spent
	time of last message	–	avg. message interval
	days active	–	std. message interval
	–	–	min. message interval
	–	–	max. message interval

 VoyTECH by Bayer and Zouzias [3] is the winning approach. It is based on gradient boosting trees (CatBoost [20]) with hand-engineered features that model the user and channel behavior without considering the content of the chat text. Since the approach does not use the textual content, the chat messages are not preprocessed. Instead, the input is modeled completely by the 26 features shown in Table 3. Some features represent superficial stylometric information that encode the message length, some interaction features encode interaction duration, but most features model the activity of users and channels. It is noteworthy that  VoyTECH uses the game as an anchor to assess the relationship between unseen users and channels, where eleven of the 26 features indicate the relationship between games, channels, and users. The authors subsample a validation dataset from the training data that is structurally similar to the test data, having a balanced distribution over user and channel activity levels as well

as a balanced number of known users. To find the optimal configuration of their model, Bayer and Zouzias carry out several experiments with varying features, differently-sized subsets of the training data, and diverse hyperparameters. They conclude that the best model on their validation dataset uses as many features and as much data as possible, as opposed to using a specific subset of the data or using only a selection of features.





 CoolStoryBob by Gärtner et al. [9] uses a feature-based gradient boosting model (XGBoost [4]), but focuses on representing the users’ texts rather than their activity or their interaction with the channels. The chat messages are preprocessed including lower-casing, removing the most and the least frequent words, common colloquial terms, stop words (NLTK’s stopword list [16]), and single-character tokens, replacing emojis and emoticons with corresponding text tokens, lemmatization (WordNet), and collapsing repetitions. The approach combines three sets of features: count vectors of the game titles, TF-IDF vectors computed from the chat messages with regard to subscription status, and hand-crafted numerical features primarily describing stylometric information as shown in Table 3. The authors subsample the original training dataset to balance the subscription status of the user-channel combinations. To find the optimal configuration of their model, the authors carry out a feature-value analysis and compare different model configurations with a five-fold cross validation.

 ItsBoshyTime exploits some shortcomings of our dataset. Since subscribers of channels can use channel-specific emotes, the usage of such emotes from the target channel reveals the ground truth about the user in question. While this approach is impractical, it provides as an interesting baseline since not all subscribed users make use of the channel-specific subscriber-only emotes available to them. To extract subscriber emotes from the training data, a dictionary of channels and their emotes was constructed via a heuristic to extract emotes from the messages in the training dataset: If a word begins with a lowercase letter and contains either a capital letter or a number, it is assumed it to be an emote. While most globally available Twitch emotes begin with a capital letter (e.g.  LUL or  PogChamp), subscriber emotes have a lower-case prefix based on the username which is usually automatically generated by Twitch.⁵ Based on this heuristic, an emote list for each channel in the training data is available. If a new user-channel combination is to be predicted, it is checked whether the channel has already been seen. If the channel is unknown, the approach defaults to predicting “not subscribed”; if an emote list for the channel is available, it is matched with the user’s list of used emotes. In case of a match the user is probably subscribed to the respective channel.

 StinkyCheese by Loures et al. [17] is based on a neural network, combining an LSTM [11] with hand-crafted features to model the verbosity, the participation, and the attendance of users towards channels. The chat messages of user-channel combinations are not preprocessed, but concatenated and fed to an LSTM layer for encoding. The resulting textual encoding is concatenated with


⁵ <https://help.twitch.tv/s/article/subscriber-emote-guide>






Table 4. Results of the competition, shown are the Precision, Recall, and F_1 as well as the runtime in H:M:S. For the random baseline, expected values are provided.



Rank	Team	Precision	Recall	F_1	Runtime
1	 VoyTECH	0.2796	0.4446	0.3433	00:07:39
2	 CoolStoryBob	0.1904	0.4341	0.2647	00:05:34
3	 ItsBoshyTime	0.4808	0.1775	0.2593	00:00:19
4	 StinkyCheese	0.0817	0.5487	0.1422	00:13:06
	Random Baseline	0.0689	0.0802	0.0741	

hand-crafted features covering all of our four categories, but each less extensively than in the other submissions. The concatenated feature vector is fed through a fully connected layer for classification. In order to handle the large dataset, the training dataset is split into chunks of 100,000 user-channel combinations, and the model trained on one of these chunks. To improve the model, the authors optimize the hyperparameters on a second 100,000 user-channel chunk.

5 Results and Discussion





The achieved performance of the participants and the random baseline are shown in Table 4.  VoyTECH outperforms the competition by a fair margin.

Relevant Features. A general trend we identify is that activity and interaction between games, channels, and users are more important than textual features. Gärtner et al. [9] reports that there is little difference in word usage between subscribed and not subscribed users. They also find that content features are of little significance in their model. In addition, the winning approach  VoyTECH does not use content features at all, but only models interaction and stylometrics when it represents activity.  StinkyCheese, which relies most on content—using an LSTM to directly incorporate the chat message contents—achieves the weakest performance. The top two approaches  VoyTECH and  CoolStoryBob explore the influence of activity groups on their performance while using very similar models. The authors of  VoyTECH additionally resampled their validation dataset based on activity.

Generalization to Unseen Users. As described in Section 3.2, 50% of the users in the test set do not appear in the training dataset, enabling an analysis whether there are any differences in prediction performance between known and new users. Table 5 shows the F_1 scores for all submissions on the test set, dependent on whether users are or are not part of the training data (*Known Users* and *New Users*, respectively). For each approach, except  StinkyCheese, users already present in the training data were more often classified correctly than new users. The drop in performance is the largest for  VoyTECH, as it relies on many user-centered features. Given only the messages of a user in the target channel

and thus missing additional information from the user’s interactions with other channels, the extracted features are less representative. Still, 🐙 VoyTECH achieves better performance than the other approaches.


Table 5. Performance difference of the submitted approaches on different subsets of the test data: Whether or not prior information about the user’s behavior in other channels is available, and dependent on channel and user activity classes.


		 VoyTECH	 CoolStoryBob	 ItsBoshyTime	 StinkyCheese	
Users	Known Users	0.3670	0.2660	0.2660	0.1410	
	New Users	0.3210	0.2630	0.2530	0.1440	
Activity	Channel	low	0.2469	0.1827	0.0505	0.1232
		normal	0.3640	0.2404	0.1718	0.1308
		high	0.4046	0.3481	0.4220	0.1738
	User	low	0.2824	0.2420	0.1781	0.0961
		normal	0.3452	0.2726	0.2689	0.1283
		high	0.3744	0.2672	0.2973	0.1716


Results by User and Channel Activity. Table 5 also shows the performance of the submitted approaches based on different channel and user activities, respectively, as defined in Section 3.2. For the most part, it can be said that, the higher the activity of a user or a channel, the better the model can predict the subscription status of a user-channel combination.

A more fine-grained activity analysis can be found in Table 6, considering all combinations of activity classes of users and channels. Again, the performance is mostly best for highly active users and channels and worst for users and channels with low activity. Most extracted features are based on the interaction of users and channels as well as their content. Having few interactions leads to less data and thus less robust features for a given user-channel combination.

Table 6. F1 scores for different user and channel activity combinations. Best values per team are written in bold.

 VoyTECH	Channel Activity			
	low	normal	high	
User Activity	low	0.184	0.298	0.351
	normal	0.272	0.331	0.412
	high	0.264	0.411	0.430

 CoolStoryBob	Channel Activity			
	low	normal	high	
User Activity	low	0.156	0.156	0.354
	normal	0.187	0.213	0.391
	high	0.189	0.272	0.320

 ItsBoshyTime	Channel Activity			
	low	normal	high	
User Activity	low	0.028	0.088	0.294
	normal	0.057	0.131	0.449
	high	0.058	0.225	0.482


 StinkyCheese	Channel Activity			
	low	normal	high	
User Activity	low	0.086	0.079	0.121
	normal	0.115	0.107	0.168
	high	0.147	0.167	0.198

Table 7. Ensembles built from the approaches submitted to the challenge. None of the ensembles outperforms the 🐡 VoyTECH approach.

Method	Precision	Recall	F ₁
🐡 VoyTECH	0.28	0.44	0.34
majority vote	0.20	0.46	0.28
“any” ensemble	0.10	0.76	0.17
“all” ensemble	0.38	0.21	0.27
“🐡 VoyTECH or else” ensemble	0.20	0.46	0.28

Ensemble Approaches. Given that the three approaches that rely on different features and classifiers (excluding 🗡 ItsBoshyTime), it is interesting to explore ensemble classification. We evaluated four different ensembles:

1. Majority vote, where users were classified as subscribed to a channel if at least two approaches say so,
2. An “any” ensemble, which classifies users as subscribed to a channel if at least one approach says so,
3. An “all” ensemble, which classifies users as subscribed to a channel if all approaches say so, and
4. A “🐡 VoyTECH or else” ensemble, which follows the classification of the best-performing approach, 🐡 VoyTECH, unless both other approaches disagree with it.

All ensembles lead to overall worse F₁ scores than the 🐡 VoyTECH approach by itself. However, as can be expected, the “any” ensemble has a notably higher recall at a lower precision, while the “all” ensemble has higher precision at lower recall. Thus, these ensembles may still be relevant when optimizing for one of these metrics. The full results for all ensembles are given in Table 7.

6 Conclusion

This paper presents the results of the ECML-PKDD ChAT Discovery Challenge 2020. It outlines the task, the datasets, the approaches, as well as the results achieved by the submissions. Our analysis of the models covers different user and channel activity groups, as well as the generalizability towards new users. We are convinced that there is potential to further improve the predictions—examples: adding message contents in the winning submission 🐡 VoyTECH for raising the model fidelity, or using ideas from 🧀 StinkyCheese for better predicting new users. While most approaches work best with highly active users and channels, the 🧑 CoolStoryBob seems to work particularly well with normally active users. Combining their features and ideas into future models may further improve the prediction quality. In addition, adding Twitch-specific features such as the sentiment of Twitch comments (e.g., extracted using the technique described

by Kobs et al. [15]) appears promising. In this challenge, the channel and user names were anonymized for privacy. However, the name of a user may give hints on the subscription status, e.g., for users who include their favorite game into their screen name. Altogether, our challenge takes a first step towards solving the task of predicting the subscription status of users at channels, giving rise to new opportunities for marketing on game streaming platforms.

Acknowledgments

We thank all participating teams for submitting their models and papers, and the ECML-PKDD organizers for hosting our shared task as a discovery challenge.

Bibliography

- [1] S. L. Anderson. Watching People Is Not a Game: Interactive Online Corporeality, Twitch.tv and Videogame Streams. *Game Studies*, 17(1), July 2017. ISSN 1604-7982. URL <http://gamestudies.org/1701/articles/anderson>.
- [2] F. Barbieri, L. Espinosa Anke, M. Ballesteros, J. Soler, and H. Saggion. Towards the Understanding of Gaming Audiences by Modeling Twitch Emotes. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 11–20, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4402. URL <http://aclweb.org/anthology/W17-4402>.
- [3] I. Bayer and A. Zouzias. Team voyTECH: User Activity Modeling with Boosting Trees. In K. Kobs, M. Potthast, M. Wiegmann, A. Zehe, B. Stein, and A. Hotho, editors, *Proceedings of the ECML-PKDD Discovery Challenge: Chat Analytics for Twitch (ChAT 2020)*, 2020.
- [4] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [5] B. C. Churchill and W. Xu. The Modem Nation: A First Study on Twitch.TV Social Structure and Player/Game Relationships. In *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, pages 223–228, Oct. 2016. doi: 10.1109/BDCloud-SocialCom-SustainCom.2016.43.
- [6] H. Davoudi, M. Zihayat, and A. An. Time-aware subscription prediction model for user acquisition in digital news media. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 135–143. SIAM, 2017.
- [7] T. Faas, L. Dombrowski, A. Young, and A. D. Miller. Watch Me Code: Programming Mentorship Communities on Twitch.tv. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–18, Nov. 2018. ISSN 2573-0142, 2573-0142. doi: 10.1145/3274319. URL <https://dl.acm.org/doi/10.1145/3274319>.
- [8] E. Gandolfi. To watch or to play, it is in the game: The game culture on Twitch.tv among performers, plays and audiences. *Journal of Gaming & Virtual Worlds*, 8(1):63–82, Mar. 2016. ISSN 1757191X, 17571928. doi: 10.1386/jgvw.8.1.63_1. URL <http://openurl.ingenta.com/content/xref?genre=article&issn=1757-191X&volume=8&issue=1&spage=63>.

- [9] M. Gärtner, A. Theissler, and M. Fernandes. Detecting Potential Subscribers on Twitch: A Text Mining Approach with XGBoost – Discovery challenge ChAT: CoolStoryBob. In K. Kobs, M. Potthast, M. Wiegmann, A. Zehe, B. Stein, and A. Hotho, editors, *Proceedings of the ECML-PKDD Discovery Challenge: Chat Analytics for Twitch (ChAT 2020)*, 2020.
- [10] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh. Wtf: The who to follow service at twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 505–514, 2013.
- [11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [12] M. R. Johnson and J. Woodcock. “It’s like the Gold Rush”: The Lives and Careers of Professional Video Game Streamers on Twitch.tv. *Information, Communication & Society*, 22(3):336–351, 2019.
- [13] M. R. Johnson and J. Woodcock. The impacts of live streaming and Twitch.tv on the video game industry. *Media, Culture & Society*, 41(5):670–688, July 2019. ISSN 0163-4437. doi: 10.1177/0163443718818363. URL <https://doi.org/10.1177/0163443718818363>. Publisher: SAGE Publications Ltd.
- [14] M. R. Johnson and J. Woodcock. “And Today’s Top Donator is”: How Live Streamers on Twitch.tv Monetize and Gamify Their Broadcasts. *Social Media + Society*, 5(4):2056305119881694, Oct. 2019. ISSN 2056-3051. doi: 10.1177/2056305119881694. URL <https://doi.org/10.1177/2056305119881694>. Publisher: SAGE Publications Ltd.
- [15] K. Kobs, A. Zehe, A. Bernstetter, J. Chibane, J. Pfister, J. Tritscher, and A. Hotho. 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):1–34, 2020.
- [16] E. Loper and S. Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.
- [17] T. Loures, G. Fernandes, F. Araújo, K. Martins, and P. Vaz de Melo. StinkyCheese: Chat-Based Model for Subscription Classification. In K. Kobs, M. Potthast, M. Wiegmann, A. Zehe, B. Stein, and A. Hotho, editors, *Proceedings of the ECML-PKDD Discovery Challenge: Chat Analytics for Twitch (ChAT 2020)*, 2020.
- [18] J. Olejniczak. A LINGUISTIC STUDY OF LANGUAGE VARIETY USED ON TWITCH.TV: DESCRIPTIVE AND CORPUS-BASED APPROACHES. page 6.
- [19] M. Potthast, T. Gollub, M. Wiegmann, and B. Stein. TIRA Integrated Research Architecture. In N. Ferro and C. Peters, editors, *Information Retrieval Evaluation in a Changing World*, The Information Retrieval Series. Springer, Sept. 2019. ISBN 978-3-030-22948-1. doi: 10.1007/978-3-030-22948-1_5.
- [20] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pages 6638–6648, 2018.
- [21] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.
- [22] J. Woodcock and M. R. Johnson. The Affective Labor and Performance of Live Streaming on Twitch.tv. *Television & New Media*, 20(8):813–823, Dec. 2019. ISSN 1527-4764. doi: 10.1177/1527476419851077. URL <https://doi.org/10.1177/1527476419851077>. Publisher: SAGE Publications.
- [23] J. Woodcock and M. R. Johnson. Live Streamers on Twitch.tv as Social Media Influencers: Chances and Challenges for Strategic Communication. *International*

Journal of Strategic Communication, 13(4):321–335, Aug. 2019. ISSN 1553-118X. doi: 10.1080/1553118X.2019.1630412. URL <https://doi.org/10.1080/1553118X.2019.1630412>. Publisher: Routledge _eprint: <https://doi.org/10.1080/1553118X.2019.1630412>.

- [24] C. Zhang and J. Liu. 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 - NOSSDAV '15*, pages 55–60, Portland, Oregon, 2015. ACM Press. ISBN 978-1-4503-3352-8. doi: 10.1145/2736084.2736091. URL <http://dl.acm.org/citation.cfm?doid=2736084.2736091>.