From PogChamps to Insights: Detecting Original Content in Twitch Chat

Jari Lindroos
University of Jyväskylä
jari.m.m.lindroos@jyu.fi

Jaakko Peltonen
Tampere University
jaakko.peltonen@tuni.fi

Tanja Välisalo
University of Jyväskylä
tanja.valisalo@jyu.fi

Raine Koskimaa
University of Jyväskylä
raine.koskimaa@jyu.fi

Ida Toivanen
University of Jyväskylä
ida.m.toivanen@jyu.fi

Abstract

The vast volume of chat messages generated during esports events on Twitch represents a valuable source of data for understanding audience behavior. However, the sheer quantity and dynamic nature of this data make manual analysis impractical. This study addresses this challenge by introducing FinTwitchBERT, a model fine-tuned to classify Twitch chat messages into four categories based on their uniqueness. Our model demonstrates the ability to distinguish between original content, repetitive messages such as emote spamming, formulaic messages, and interactive commands chat participants use to interact with channel bots. Pre-trained on over 18 million Finnish Twitch chat messages and utilizing a combination of semi-supervised learning and iterative pseudo-labeling with human-in-the-loop validation, FinTwitchBERT achieves 97.42% accuracy on a test set of unseen chat messages with a limited initial dataset of only 7,529 manually annotated messages.

Keywords: social media analysis, Twitch, chat, machine learning, natural language processing

1. Introduction

The esports public image is saturated with images of big arena games with huge displays, pyrotechnics, and cheering audience. This is only the tip of the iceberg, however, and the main form of esports is online tournaments, where the teams play in their own studios, and the audience follows the game through the online streaming platform. Even in the case of live events, the big LAN tournaments, there is much more audience following the live streams than in the arena. When the audience is following the streams, most often in the Twitch.tv site, they also have a chance to attend the accompanying chat discussion. In the chat, the audience is commenting on the events taking place in the game, they cheer for their favourite teams and players, give feedback to the commentators, engage in discussions with other audience members, and often simply send simple messages such as ”gg” (good game) to show that they are present.

It is important to look at these chat discussions in detail to better understand the behaviors, preferences, and attitudes of the esports audience. Researchers can analyze chat messages to gain a better understanding of what viewers are interested in, what they dislike, and how they engage with the content, the stream provider and each other.

The challenge with chat messages as research data is its massive quantity. One game may yield thousands of messages, and there may be several simultaneous channels each with their own chat. A tournament often consists of dozens of games, and there are constantly several tournaments under way. The data quickly grows into hundreds of thousands, even millions of messages. To gain a more robust understanding of the chat contents and dynamics, automated tools are required.

We propose that one approach to both deal with the vast volume of data and identify important aspects of viewers' chat behaviour is to identify repeated expressions and the patterns of their appearance. It is the nature of the Twitch chat, that a number of more or less conventionalized expressions, emotes and memes are used regularly, and often the messages are just replicating these (”LUL”, ”KEKW” etc.), or they may be chat-specific commands directed towards the platform itself (maps!) On the other hand, there are also messages, where a person is writing an authentic message to comment on the game, or to engage other

URI: https://hdl.handle.net/10125/109148
978-0-9981331-8-8

audience members. Identifying different types of chat content allows researchers to analyze the frequency of their appearence and their communicative context, such as what was posted in the chat simultaneously and what was occurring in the livestream.

Different research topics necessitate different types of chat content. For example, interest towards diffusion of internet slang could make it important to recognize the repetitive contents of the chat, whereas the study of audience attitudes towards the casters or the organization behind the stream could entail discourse analysis with a strong focus on unique content. Research on forms of interaction between chat participants as they direct messages to each other could include analysing how unique and non-unique content are used as part of the communicative repertoires of the discussants.

Machine learning based text classification methods are important in turning unstructured text into structured knowledge. Implementing accurate text classification techniques heavily relies on domain-specific annotations to build massive machine learning models, and although these models can show superior performance, the lack of training data and the cost and time-consuming nature of manually annotating the data can pose a significant bottleneck in developing these models in practice.

To address these challenges, we propose an approach that combines semi-supervised learning with iterative pseudo-labeling using human-in-the-loop validation for feedback to effectively classify Twitch chat messages into four distinct categories: 1) unique, 2) non-unique, 3) essentially non-unique but human-written, and 4) commands and replies. Through our research we attempt to answer the following research questions:

1. Can Twitch livestream chat messages be distinguished by the level of uniqueness?

2. Do pseudo labeling based methods provide good enough annotation quality, so that pseudo labeled training data do not deteriorate model performance?

2. Previous research

2.1. Machine learning for Twitch data

Recent studies involve ever more machine learning and deep learning methods applied to massive chat data. In the case of Twitch, Kim et al., 2019 proposed a framework where the sentiments of chat users are tracked to identify subscribers of the particular channel only based on their messages. The researchers tried distinguishing subscribers from non-subscribers in sentiment-level by using a series of t-tests and found that subscribers are leaning into an analytical style in conversation while non-subscribers have a more emotional and perceptual vocabulary in their speech. For extracting sentiment features from text, Linguistic Inquiry and Word Count (LIWC) was used. More recently, a study by Dolin et al., 2021 proposed a new baseline for sentiment analysis tailored to Twitch data. According to the paper, Twitch chat has a set of emotes, up to 8.06 million, that users incorporate in their messages. These emotes are not clearly defined in the dictionary, as regular vocabulary is, which speaks for developing Twitch-specific sentiment analysis. In the study word embeddings along with k-nearest neighbors were applied in an unsupervised fashion to incorporate new vocabulary into existing models (Dolin et al., 2021). This is important, since modern language models are typically trained with language that might not include terminology (i.e., emotes) specific to subcultures, making the model miss their contextual meanings altogether or not be aware of the important connotations of the terminology. With the framework presented, a pseudo-dictionary of emotes can be generated automatically to support future work on Twitch data. The need for research about Twitch emotes, specifically predicting them, has also been noted in previous literature (Barbieri et al., 2017). In another study, researchers developed two unsupervised lexicon-based approaches and a weakly supervised neural network-based classifier for sentiment analysis on chat messages during Twitch livestreams. The methods were able to generate sentiment trajectories during livestreams, which correlated well with specific topics in the given stream (Kobs, Zehe, et al., 2020).

In addition to sentiment analysis, machine learning has been used to predict viewer engagement. Melhart et al., 2020 used shallow artificial neural networks to predict low or high engagement based on gameplay events in PlayerUnknown's Battlegrounds (PUBG) livestreams. Their analysis encompasses both chat logs and game telemetry data, with specific gameplay features (e.g., player health status, in-game choices, and map location). The results demonstrate the possibility of accurately predicting continuous viewer engagement based exclusively on key gameplay events.

While machine learning methods may represent more traditional artificial intelligence research, there has recently been a surge of research that is specifically founded in deep learning. There are studies that focus on highlight detection (Liaw & Dai, 2020; Song et al., 2021) and predicting chat participants' subscription status based on their chat messages (Loures et al., 2020). All of the models share a similar base architecture: long short-term memory, or LSTM in short. For example,

Liaw and Dai, 2020 explored the use of chat messages as an alternative way to identify highlight-moments within livestream videos. The authors proposed a LSTM-based model with three features to capture and interpret the chat participants' reactions. The first feature was based on the frequency of the messages within a certain time window. The second feature was based on the diversity of chat messages, describing how the chat participants tend to communicate in a similar way during a highlight moment. The last feature was based on the semantic information to understand the emotion behind the chat participants' message content. The results of the model demonstrated that utilizing chat-based features are more stable than utilizing vision-based features.

Similarly focusing on highlights, Song et al., 2021 proposed methods for automating the detection of exciting or engaging moments in livestreams. Their methods explored the relationship between chat participants' reactions in the chat messages and the events occurring on screen to identify the category of the potential highlight (e.g., victories, funny situations). The authors proposed a model named Multimodal Detection with INTerpretability (MINT), which integrates distinct deep learning models to analyze chat, video, and frequency-based features. The researchers suggested including transformer-based architectures, such as BERT (Devlin et al., 2018), for processing chat participants messages to imporove the performance of the model.

Deep learning has also been used to predict the level of user engagement with the channel, based on chat data. Loures et al., 2020 proposed a so-called LSTM-based "StinkyCheese" model for classifying chat participants subscription status based on their chat messages in the ECML-PKDD 2020 Discovery Challenge (Kobs, Potthast, et al., 2020). The authors combined both text representations of the chat messages with additional descriptive features, based on verbosity (e.g., the total number of messages and the average message length), attendance (e.g., the time between the chat participants last and first messages, or the average time between messages sent by a chat participant) and participation (e.g., how many channels a certain user participates in and the number of users participating in the current stream).

Studies implementing BERT models (e.g., Z. Gao et al., 2020), or other transformer models, for Twitch chat data are few and far between, and some studies even suggested that future research should have more of a focus on BERT models (Kim et al., 2019; Song et al., 2021). However, some research has been conducted using BERT models with other livestreaming platforms (see, e.g., J. Gao et al., 2023). Additionally, multimodality has also been taken into account in machine learning based esports research. In a recent study (L. Zhang & Wang, 2023), researchers proposed several end-to-end models to learn joint representations of multiple modalities (visual, auditory, and textual) from Counter-Strike: Global Offensive (Valve 2012) (CS:GO for short) gameplay videos from Twitch. The models were able to assess players' gaming skills from videos with comparable accuracy.

2.2. Transformer based architectures

Transformer-based architectures (Vaswani et al., 2017), like BERT (Devlin et al., 2018), which were originally introduced for machine translation tasks, have revolutionized natural language processing (NLP) by achieving state-of-the-art performance in a wide range of natural language processing tasks. Their ability to capture long-range dependencies between words and phrases has made them particularly well-suited for tasks like text summarization, question answering, and natural language generation, outperforming traditional convolutional neural networks. One of the building blocks in BERT models is the masked language modeling (MLM) objective. In MLM a word, or more precisely a token, is understood through its context by including information bidirectionally. Formally, a token z_t is replaced by a mask for the model to predict it using previous and following tokens $\mathbf{Z}_{\setminus t} := (z_1, ..., z_{t-1}, z_{t+1}, ..., z_{|\mathbf{Z}|})$ (Salazar et al., 2019).

Although multilingual BERT-based models have shown promise in various languages, language-specific models such as Finnish BERT (Virtanen et al., 2019) have been shown to outperform multilingual BERT-based models for Finnish-specific natural language processing tasks. Virtanen et al., 2019 highlight the importance of tailoring language models to specific languages, especially for lower-resourced languages like Finnish, for better performance and capturing the linguistic nuances of each language more accurately.

2.3. Semi-supervised learning and iterative pseudo labeling

The increasing volume of data currently presents a challenge for natural language processing methods, especially in the field of text classification which revolves around assigning labels to text based on their content. Traditional approaches tend to rely on models which have been trained on manually annotated large datasets. This approach however tends to be expensive and time-consuming.

To address this issue, semi-supervised learning

(SSL) has shown success by utilizing both labeled data and unlabeled data during training a machine learning model (Duarte and Berton, 2023). One particularly effective SSL technique is pseudo labeling (Lee et al., 2013), in which the model is first trained on labeled data and it is then used to predict labels for the unlabeled data. These newly predicted pseudo labels can then be added to the training set, further expanding the amount of available labeled data. More precisely, pseudo labeling can be defined as follows (Arazo et al., 2020). Let us denote a training dataset as $\mathcal{T}$ that is the size of N samples. From the training dataset samples are randomly picked to form a labeled subset $\mathcal{T}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$, where $y_i \in \{0,1\}^C$ is one-hot encoding label for C classes corresponding to x_i. The remaining samples form a subset of unlabeled samples, which is defined as $\mathcal{T}_u = \{x_i\}_{i=1}^{N_u}$, so that $N = N_l + N_u$.

Pseudo labeling has been shown to improve performance in text classification benchmarks without the explicit need for in-domain unlabeled data (Du et al., 2020). In our work we leverage both SSL and iterative pseudo labeling (IPL) based on human-in-the-loop (HITL) validation strategy. Studies utilizing SSL based IPL frameworks with HITL validation (Wang et al., 2022) and without HITL (Xu et al., 2020, Z. Zhang et al., 2022, Likhomanenko et al., 2020, Benato et al., 2021, Li et al., 2023, Huang et al., 2022) have shown accurate model performance and robustness when trained on a limited amount of labeled data.

3. Methods

3.1. Unique content detection

Twitch chat messages contain a massive volume of real-time conversations within the broad gaming and livestreaming community, offering a unique opportunity to capture the dynamic and diverse language patterns of online audience engagement. However, utilizing this data for text classification based on chat contents also presents unique challenges due to the unique characteristics of Twitch chat messages.

Transformer based models, like BERT, have mostly been trained on general-purpose text, and may not effectively recognize these patterns or interpret the constantly evolving culture and context of chat conversations in the livestreaming platforms. To address these challenges, we propose an approach that combines semi-supervised learning (Zhu and Goldberg, 2009) with iterative pseudo-labeling using human-in-the-loop validation (Wang et al., 2022) for feedback to effectively classify Twitch chat messages into four distinct categories (examples in Figure 1). The classification scheme was developed through rounds of data annotation and discussions by the authors. The end result included the four following classes by uniqueness:

- **Unique** (category 0). This category consists of unique messages devised by their authors. These messages stand out as original, informative, and directly relevant to the stream content. Such messages are not direct copies of any other messages, and are typed in by Twitch viewers to comment on something specific that adds value to the chat interaction. However, there can still be repetition in unique messages and they can comment on the same topic or event as other participants, expressing a common sentiment. An example of such a message is “The major finals really ought to be BO5 tbh; G2 would be strong but have to play shorthanded in practice as that aleksi [won’t] get anyone”, where the writer comments on their preferred teams for upcoming games.

- **Non-unique** (category 1). This category includes repetitive content, meaning that the same message is posted by many different viewers. In practice, the category includes emotes, emojis, game slang and copypasta. Emojis are smileys or other images included in the Unicode standard. The Twitch platform also has its own emotes which can be common for the whole platform (e.g., OMEGALUL, Kappa, PogChamp – see images of these emotes in Figure 1 under class ”Non-unique”, listed as global Twitch emotes), or tailored to a particular channel. Common expressions of game slang are abbreviations, such as ‘gg’ (good game) or ’wp’ (well played). While these messages might not directly contribute to the stream’s discussion, they are still a significant part of the Twitch chat culture and integral in establishing a shared communication style among viewers.

- **Essentially non-unique but human-written** (category 2). The category consists of formulaic content, which is based on variations of certain patterns. The messages in this category can be reactions (e.g., Yesyesyes, oh wowww), cheers and chants (e.g., ez4ence, LETS GO G2) or short-spoken expressions that add a personal touch to the chat and contribute to the overall atmosphere of the stream. Messages falling into this category are not unique in terms of content but are “human-written“ in the sense that there is more variety in the expression and context,

reflecting a range of individual responses that are less standardized but still follow recognizable patterns in how they are formulated.

- **Commands and replies** (category 3). Messages in this category are either commands (e.g., "!casters", "!maps") written by the participants to a chatbot (e.g., Nightbot, Moobot), or responses from the bot (e.g., "All games are BO1 for today"). These messages can be used, for example, to ask for information or participate in a raffle. These messages are functional and play a role in engaging viewers with the stream elements beyond mere conversation.

The aim was to develop a model that could detect the uniqueness category of a message based only on the text of the message itself, without any contextual information such as the prevalence of the message or known characteristics of the commenter or the match. Such a model can be used during esports events to detect potential unique or non-unique comments as they occur, without needing to retain any earlier chat data.

Figure 1. Examples of chat contents by unique classes.

3.2. Data

For our dataset, we collected a large amount of chat messages from the Twitch livestreaming platform. A total of 30,486,995 messages was collected from late 2022 to February 2023 from 64,108 available past broadcasts and clips of Finnish Twitch channels out of which 18,556,229 were not duplicate (see Table 2 for final label distribution). The dataset contains messages from 3,281 different categories and games to ensure a wide range of language and communication styles specific to Finnish Twitch streams. The collected Twitch stream data contains text mostly in Finnish, but also includes other languages, such as English and Swedish.

To ensure a diverse representation of the language in Twitch chat, we perform the following preprocessing steps. We remove exact duplicate messages to reduce bias towards frequently repeated phrases. Additionally, we convert Unicode emojis to their equivalent textual descriptors (e.g., ☺ → :smiling_face:). This standardization helps the model better understand the language patterns within Twitch chat.

Algorithm 1 IPL with Human-in-the-Loop (HITL)

Input: L: Labeled dataset U: Unlabeled dataset M: Model T: Confidence threshold N: Number of iterations
Output: Trained model M

Train initial model M on L
for $i = 1$ to N **do**
  ▷ Pseudo-Labeling
  $\hat{y}_j \leftarrow \text{argmax}\, M(x_j) \;\; \forall x_j \in U$
  ▷ Confidence Filtering
  $U' \leftarrow \{(x_j, \hat{y}_j) \mid \max M(x_j) \geq T\}$
  ▷ Human-in-the-Loop Verification
  $U'' \leftarrow \text{HumanVerification}(U')$
  ▷ Augmentation
  $L' \leftarrow L \cup U''$
  ▷ Retraining
  Retrain model M on L'
end for

3.3. Iterative pseudo-labeling

Iterative pseudo-labeling (IPL) with human-in-the-loop was utilized to annotate the data into the four classes (Algorithm 1). The pseudo-labeling process started by randomly selecting day 7 of the PGL Major 2022 CS:GO Tournament from the channel 'yleeurheilu' of the Finnish national broadcasting company YLE as the initial dataset for annotation. The chosen data was divided into four parts, with each part sampled to ensure an even distribution of messages across the entire duration of the stream. Each part was then manually annotated by a different annotator, all of whom have experience in understanding Twitch chat. To ensure consistency in the first phase of annotations, any observed discrepancies in the annotations were discussed and addressed among the annotators before the four parts were combined.

Then, we iteratively pseudo-labeled three additional

Table 1. Stages of iterative pseudo-labeling

Stage	Total messages	Description
1	7,529	Manual annotation of day 7 of PGL Majors ('yleeurheilu' channel)
2	25,903	Iterative pseudo-labeling of day 1, day 2 and day 3 ('yleeurheilu' channel)
3	107,920	Iterative labeling for the rest of the tournament for both channels 'yleeurheilu' and 'pelaajatcom'
4	112,646	Expand data with 4,726 custom emotes
5	1,004,726	Iterative pseudo-label subset of 1 million messages
6	3,004,726	Iterative pseudo-label 2 million more messages
7	5,004,726	Iterative pseudo-label 2 million more messages
Final	5,004,726	A total of over 5 million messages were used to train the final model

Table 2. Label distribution for the training and validation datasets.

Category	Training	Validation
Unique	3 318 547	368 403
Non-unique	162 582	18 343
Essentially non-unique but human-written	870 480	96 862
Commands and replies	152 644	16 865

days of data from 'yleeurheilu', which were manually checked and corrected utilizing human feedback. This cycle was continued until we had labeled the data from the entire tournament – 12 days of data from 'yleeurheilu' and 6 days from 'pelaajatcom' for a total of almost 108,000 messages. Then, we expanded the range of emotes included in the dataset with an external dataset of 4,726 custom 7TV emotes to introduce a larger variety of emotes, beyond the original Twitch emotes. We then continued the series of iterative pseudo-labeling on a larger-scale data, for which we first split the pre-training collection into five 1-million unlabeled subsets. To enhance the generalization of our model, we iteratively increased the labeled data size, starting from 1,047 million unique messages, then progressively scaling up to 3,047 million and finally up to 5,047 million unique messages. The final model involved six stages of iterative pseudo-labeling and human-in-the-loop analysis was conducted after each iteration of pseudo-labeling to validate the accuracy and consistency of the labels. A summary of all the stages in the data annotation process can be viewed in the Table 1.

3.4. Modeling

We chose to use a deep neural transformer network based classification scheme, in order to take into account the various forms that messages could be typed in with variant spellings, word order, and added embellishments (e.g., emojis in between words or letters without changing the unique or non-unique nature of the comment. We used a transformer-based classification architecture to utilize its self-attention mechanisms. This allowed the model to dynamically weigh the importance of different words and phrases within a single chat comment, promoting a deeper semantic understanding necessary for the unique/non-unique classification task.

First, we conducted in-domain pretraining on the FinBERT-base model (Virtanen et al., 2019) using a large dataset of 18,556,229 unique messages collected from Finnish Twitch channels past broadcasts. The pretraining was done by using the masked-language-modeling (MLM) objective (Devlin et al., 2018). This allowed the model to learn the vocabulary, linguistic nuances and domain-specific patterns going on in the Finnish Twitch chat (to model the dataset). No uniqueness labels are available for this large data set of 18,5 million messages. For the pretraining we used the following hyperparameters: maximum sequence length of 200, batch size of 128 for training and validation, AdamW optimizer with initial learning rate of 1e-6, linear scheduler with no warmup steps and a weight decay of 0.01, and masked language modeling with a probability of 0.15. We pretrained the model for a total of two epochs.

After pretraining, we focused on supervised training of the model. We first used a manual annotation process to first create a labeled dataset for the uniqueness-specific classification task. We then utilized iterative pseudo labeling to augment a larger unlabeled dataset with human feedback on the labels, creating an even larger annotated dataset to finalize the supervised model training and to create a final model. The final model was trained with over 5 million annotated data samples (see Table 2). For the supervised training of the final model, we used the following training configuration/setup. We trained the pretrained FinBERT-base model in Pytorch 1.8.1+cu102 for a total of 5 epochs using a batch size of 126 and a sequence length of 250 on 3x Tesla P100 GPUs. We employ a learning rate scheduler with warmup and cosine decay, starting from an initial learning rate of 5e-4 and reducing it to 0 on the final epoch. The optimizer used was AdamW with different learning rates for different layers of the model: 3e-5 for the fully connected-layer, 3e-5

Table 3. Performance of FinTwitchBERT.

Split	Accuracy	F1	Recall	Precision
Validation	0.97132	0.98150	0.97540	0.98782
Test	0.97423	0.97422	0.97423	0.97450

for the pooler layer, and 1e-5 for the rest of the models layers. We also utilized WeightedLayerPooling, where the token embeddings are weighted mean of their hidden layer representations, and multi-sample dropout (Inoue, 2019) to speed up training and improve generalization.

4. Results

This section presents the experimental results of our research, in which we evaluate the performance of our FinTwitchBERT model for classifying Twitch messages by uniqueness categories on both the validation set and the curated test set. Additionally, we provide insights on the performance, and highlight certain challenges faced in classifying specific uniqueness categories.

For evaluating the model's capability to generalize to unseen data, we curated the test set to ensure a diverse quality and representation of different types of chat messages, including messages from different games, streamers, and Twitch categories. The test set consists of 10,440 deduplicated messages collected from Finnish Twitch categories during the first week of October 2023, including tournament matches from games such as CS:GO or streamer gameplay from The Sims 4. Also categories like Just Chatting and Slots are included to represent livestream chat data not specific to only gameplay-based streams. This allows us to assess how well the model generalizes to new data and performs on chat messages from a variety of sources.

The final model achieved 97.1% accuracy and 98.1% F1 score on the validation set, and 97.4% accuracy and 97.4% F1 score on the test set (Table 3). The consistent performance across both sets not only indicates the model's capacity to generalize effectively to unseen data but also supports our second research question that the pseudo-labeling process generated sufficient-quality annotations, thus not deteriorating the model's capability to learn from the pseudo-labeled data.

The high accuracy and F1 scores of the FinTwitchBERT model could be partially attributed to the careful annotation process employed. This process combined iterative pseudo-labeling with a human-in-the-loop approach, allowing for the benefits of both machine-generated annotations and the corrective input of a human annotators. While multiple annotators were initially involved in the labeling process, a single, experienced annotator, familiar with Twitch livestreams and their unique chat dynamics, participated in reviewing all of the annotations at every step for consistency and quality of the labels. This approach likely improved the training dataset's reliability while also reducing the need for extensive resources for the annotation process.

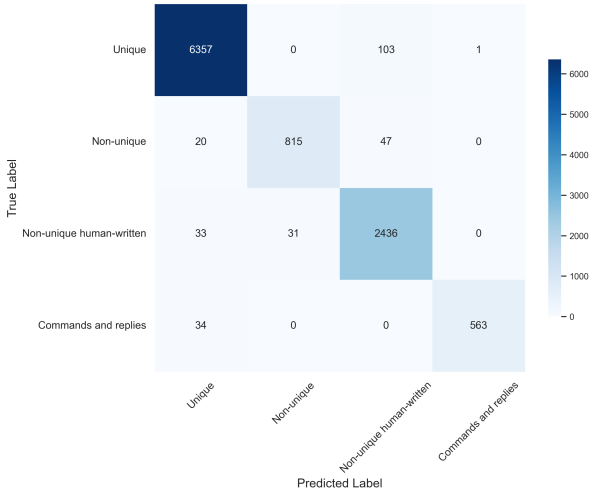

Figure 2. Confusion matrix of test set predictions.

The definition of the uniqueness classes are also designed so that they are mutually exclusive. There is no overlap between the classes.

While the model showed great overall performance, there were specific challenges in classifying certain types of content, particularly within the different classes. Analysis of the Figure 2 confusion matrix reveals that the most frequent misclassifications for the unique-class were incorrectly categorized as non-unique but human-written. While correctly classified unique messages have a mean token length of 43.26 (std 34.09), unique messages incorrectly classified as non-unique but human-written exhibit a shorter mean length of 20.6 (std 10.34). Interestingly, the single misclassification into the "commands and replies" class represents a relatively long text sample (*"Could I get one pair of adidas pants with four stripes thank you?"*).

Correctly classified unique messages exhibit a wide range of token lengths (min: 7, max: 553). In contrast, unique messages which were incorrectly classified as either non-unique but human-written or commands and replies fall within a narrower range of 9 to 70 tokens.

For the samples belonging to the class non-unique messages, there are misclassifications to the classes unique and non-unique but human written. This could be due to the messages containing unseen custom emotes or subscription emotes from small streamers and specific one-worded terms posed challenges. For

example, a message containing just the number (”250”, which in Chinese slang can be used as an insult) was classified as non-unique but human written, despite the human labeling being ”non-unique”. Most of the misclassifications happening seemed to occur for more rare subscription-based emotes from smaller audience Finnish streamers where the emotes are used once or multiple times in a single message (Figure 3). Furthermore, correctly classified non-unique samples had an average length of 20.90 tokens (std 39.99, min: 1, max: 447). Misclassified samples exhibited varying lengths: those labeled as unique averaged 67.5 tokens (std 32.96, min: 23, max: 155), while those labeled as non-unique but human-written averaged 20.21 tokens (std 21.64, min: 1, max: 107).

The non-unique but human written messages tend to be misclassified as unique or non-unique. Mostly short one-word messages, such as ”stormforged”, ”wahuu” and ”Otto”, were wrongly classified as non-unique. Interestingly, these misclassified messages exhibited similar average lengths (mean: 15.67, std: 26.21) to correctly classified non-unique but human-written messages (mean: 13.66, std: 11.99). However, messages incorrectly classified as unique tended to be longer (mean: 27.33, std: 29.95).

Incorrectly classified messages actually belonging to the commands and replies class tend to be misclassified as unique. Certain types of unseen, less common commands, such as ”!ask xxxx” (where the xxxx resembles a passage of text which could be seen as unique). Certain channel advertisements or system messages were also misclassified as unique due to the contents of the message resembling uniqueness. Correctly classified commands and replies had an average length of 40.16 tokens (std 36.58, min: 2, max: 278). In contrast, misclassified samples labeled as unique were significantly longer (mean: 84.97, std 66.92, min: 26, max: 301).

It is noteworthy that even human annotators faced certain challenges in distinguishing messages between unique and essentially non-unique but human-written, highlighting a certain complexity of the labels for the uniqueness classification task.

Message	PREDICTED	ACTUAL	Description
Check out the Gigantti Gaming offers of the month! https://pelaajat.com/out/gigantti1	Unique	Commands and replies	Advertisement
!ask is it worth taking my bike with me if I'm just walking it	Unique	Commands and replies	Command which resemblances an unique message
AAAAAH that badge!!	Essentially non-unique but human-written	Unique	Context dependency, could be either
cha cha cha	Non-unique	Essentially non-unique but human-written	Emoji of resemblance to Kääriä from the Eurovision
You will receive your next merit badge after 15376 bits!	Unique	Commands and replies	Channel chatbot message
smallstreamerEmote1 smallstreamerEmote2 smallstreamerEmote3	Essentially non-unique but human-written	Non-unique	Subscription emotes from smaller streamers

Figure 3. Examples of wrong predictions.

5. Discussion and conclusion

In this study we present an approach for classifying Twitch chat messages into four distinct classes to capture the complexity and diversity of communication on the platform. While a simple ’unique’ vs. ’non-unique’ split might capture broad differences, we feel it overlooks the nuanced roles these messages play in the chat. Drawing from cultural and media studies perspectives, our four-category system distinguishes between repetitive messages, formulaic human-written content, and functional commands, each serving different communicative functions. This approach offers a more comprehensive understanding of the rich communication patterns present in the livestream chats. During model development, we made a deliberate choice to focus on the message contents specific to Twitch and while the method was tailored with Twitch in mind, it could still be applied to other large online cultural communities that share similar chat message characteristics.

Our proposed method uses transformer-based models like BERT in Finnish language combined with semi-supervised learning utilizing iterative pseudo-labeling and human-in-the-loop validation for feedback, demonstrating exceptional results in terms of both accuracy and efficiency, achieving an accuracy of 97.42% on a test dataset of unseen chat messages with just originally 7,529 manually annotated messages.

The model’s performance demonstrates its ability to effectively capture the nuanced characteristics of Twitch chat across the four categories of chat messages. This performance is further supported by the iterative pseudo-labeling process, which made it possible to effectively scale the size of the training data while maintaining annotation quality via human-in-the-loop, thus reducing the need for extensive manual annotation

resources. This capability to effectively analyze and distinguish between different types of chat messages is important for understanding the dynamics and social interactions within Twitch chat, providing valuable insights for both researchers and the streaming community.

While our method demonstrated overall good efficiency in capturing the nuanced differences between the four categories of chat messages, it also faced challenges in certain cases. For instance, differentiating between unique messages and essentially non-unique but human-written messages proved challenging in certain cases where the length of the message would be shorter for unique messages or longer for non-unique but human-written messages. Additionally messages, that contain unseen subscription emotes from smaller streamers or specific one-word terms could be misclassified as either unique or non-unique but human-written as opposed to non-unique.

Another challenge involved classifying certain types of commands and advertisement messages that share similarities with unique type of messages, making them difficult to categorize accurately without added information about the usernames of known channel bots.

One source of misclassification appears to be related to message length. Our analysis revealed that shorter unique messages were in some cases misclassified as non-unique but human-written, while longer messages typically labeled as non-unique but human-written were sometimes misclassified as unique. Further research is needed to investigate whether the model is overly reliant on message length as a distinguishing factor. Unique messages may exhibit greater syntactic complexity or a wider range of vocabulary, while non-unique messages might rely more on repetitive phrases, emotes or livestream slang specific to a particular streamer's community. Analyzing linguistic features beyond simple length could help the model refine its classification.

The findings also highlight some of the inherent challenges in classifying Twitch chat messages, even for human annotators. Additionally the classification task is further complicated by the constantly evolving linguistic features of Twitch subcultures. New emotes, copypastas, and community-specific communication styles emerge and gain popularity at a rapid pace. This continuous evolution means that models trained on these static datasets will struggle to keep up with the dynamic nature of communication on the platform. As new linguistic features become prevalent, misclassifications are more likely to increase unless the model can be continually adapted or retrained with newer data that reflects the current chat environment.

While this paper primarily focuses on model development, leaving detailed analysis of messaging behavior for future work, the classification system we propose lays the groundwork for such analyses. This type of classification is a first step in identifying different discursive strategies within chat messages, which can ultimately provide deeper insights into the behavior, preferences, and attitudes of the livestreaming audience.

6. Acknowledgements

This research was supported by the FIN-CLARIAH infrastructure project (The Research Council of Finland 358726).

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., & McGuinness, K. (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. *2020 International joint conference on neural networks (IJCNN)*, 1–8.

Barbieri, F., Espinosa-Anke, L., Ballesteros, M., Saggion, H., et al. (2017). Towards the understanding of gaming audiences by modeling twitch emotes. *Third Workshop on Noisy User-generated Text (W-NUT 2017); 2017 Sep 7; Copenhagen, Denmark. Stroudsburg (PA): ACL; 2017. p. 11-20.*

Benato, B. C., Telea, A. C., & Falcão, A. X. (2021). Iterative pseudo-labeling with deep feature annotation and confidence-based sampling. *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 192–198.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dolin, P., d'Hauthuille, L., & Vattani, A. (2021). Feelsgoodman: Inferring semantics of twitch neologisms. *arXiv preprint arXiv:2108.08411*.

Du, J., Grave, E., Gunel, B., Chaudhary, V., Celebi, O., Auli, M., Stoyanov, V., & Conneau, A. (2020). Self-training improves pre-training for natural language understanding. *arXiv preprint arXiv:2010.02194*.

Duarte, J. M., & Berton, L. (2023). A review of semi-supervised learning for text classification. *Artificial Intelligence Review*, *56*(9), 9401–9469.

Gao, J., Lian, Y., Zhou, Z., Fu, Y., & Wang, B. (2023). Livechat: A large-scale personalized dialogue

dataset automatically constructed from live streaming. *arXiv preprint arXiv:2306.08401*.

Gao, Z., Yada, S., Wakamiya, S., & Aramaki, E. (2020). Offensive language detection on video live streaming chat. *Proceedings of the 28th international conference on computational linguistics*, 1936–1940.

Huang, Z., Wang, H., Ye, J., Niu, J., Tu, C., Yang, Y., Du, S., Deng, Z., Gu, L., & He, J. (2022). Revisiting nnu-net for iterative pseudo labeling and efficient sliding window inference. In *Miccai challenge on fast and low-resource semi-supervised abdominal organ segmentation* (pp. 178–189). Springer.

Inoue, H. (2019). Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*.

Kim, J., Bae, K., Park, E., & del Pobil, A. P. (2019). Who will subscribe to my streaming channel? the case of twitch. *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, 247–251.

Kobs, K., Potthast, M., Wiegmann, M., Zehe, A., Stein, B., & Hotho, A. (2020). Towards predicting the subscription status of twitch.tv users — ecml-pkdd chat discovery challenge 2020. *Proceedings of ECML-PKDD 2020 ChAT Discovery Challenge*.

Kobs, K., Zehe, A., Bernstetter, A., Chibane, J., Pfister, J., Tritscher, J., & Hotho, A. (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), 1–34.

Lee, D.-H., et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning, ICML*, *3*(2), 896.

Li, T., Meng, Q., & Sun, Y. (2023). Improved noisy iterative pseudo-labeling for semi-supervised speech recognition. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 167–173.

Liaw, C.-M., & Dai, B.-R. (2020). Live stream highlight detection using chat messages. *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, 328–332.

Likhomanenko, T., Xu, Q., Kahn, J., Synnaeve, G., & Collobert, R. (2020). Slimipl: Language-model-free iterative pseudo-labeling. *arXiv preprint arXiv:2010.11524*.

Loures, T. C., Fernandes, G. L., Araújo, F. G., Martins, K. S., & de Melo, P. O. V. (2020). Stinkycheese: Chat-based model for subscription classification. *ChAT@ PKDD/ECML*.

Melhart, D., Gravina, D., & Yannakakis, G. N. (2020). Moment-to-moment engagement prediction through the eyes of the observer: Pubg streaming on twitch. *Proceedings of the 15th International Conference on the Foundations of Digital Games*, 1–10.

Salazar, J., Liang, D., Nguyen, T. Q., & Kirchhoff, K. (2019). Masked language model scoring. *arXiv preprint arXiv:1910.14659*.

Song, H., Park, K., & Cha, M. (2021). Finding epic moments in live content through deep learning on collective decisions. *EPJ Data Science*, *10*(1), 43.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F., & Pyysalo, S. (2019). Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.

Wang, M., Valmianski, I., Amatriain, X., & Kannan, A. (2022). Learning functional sections in medical conversations: Iterative pseudo-labeling and human-in-the-loop approach. *arXiv preprint arXiv:2210.02658*.

Xu, Q., Likhomanenko, T., Kahn, J., Hannun, A., Synnaeve, G., & Collobert, R. (2020). Iterative pseudo-labeling for speech recognition. *arXiv preprint arXiv:2005.09267*.

Zhang, L., & Wang, W. (2023). Multi-modal machine learning for assessing gaming skills in online streaming: A case study with cs: Go. *arXiv preprint arXiv:2307.12236*.

Zhang, Z., Bansal, N., Cai, C., Ji, P., Yan, Q., Xu, X., & Xu, Y. (2022). Clip-flow: Contrastive learning by semi-supervised iterative pseudo labeling for optical flow estimation. *arXiv preprint arXiv:2210.14383*.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*. https : / / api . semanticscholar . org / CorpusID:40097546