Computer Speech and Language 88 (2024) 101651

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Applying machine learning to assess emotional reactions to video game content streamed on Spanish Twitch channels

Noemí Merayo [a,*], Rosalía Cotelo [b], Rocío Carratalá-Sáez [c], Francisco J. Andújar [d]

[a] Dept. Teoría de la Señal y Comunicaciones e Ingeniería Telemática, Universidad de Valladolid, Paseo de Belén 15, Valladolid, 47011, Spain
[b] Dept. de Filologías y su Didáctica, Universidad Autónoma de Madrid, Madrid, 28049, Spain
[c] Dpto. de Ingeniería y Ciencia de los Computadores, Universitat Jaume I, Castellón de la Plana, 12071, Spain
[d] Dpto. de Informática, Universidad de Valladolid, Paseo de Belén 15, Valladolid, 47011, Spain

ARTICLE INFO

Dataset link: https://github.com/noemeralv/Corpus-Twitch-Videogames

Keywords:
Emotional response
Corpus
Machine learning
Twitch
Video games

ABSTRACT

This research explores for the first time the application of machine learning to detect emotional responses in video game streaming channels, specifically on Twitch, the most widely used platform for broadcasting content. Analyzing sentiment in gaming contexts is difficult due to the brevity of messages, the lack of context, and the use of informal language, which is exacerbated in the gaming environment by slang, abbreviations, memes, and jargon. First, a novel Spanish corpus was created from chat messages on Spanish video game Twitch channels, manually labeled for polarity and emotions. It is noteworthy as the first Spanish corpus for analyzing social responses on Twitch. Secondly, machine learning algorithms were used to classify polarity and emotions offering promising evaluations. The methodology followed in this work consists of three main steps: (1) Extracting Twitch chat messages from Spanish streamers' channels related to gaming events and gameplays; (2) Processing and selecting the messages to form the corpus and manually annotating polarity and emotions; and (3) Applying machine learning models to detect polarity and emotions in the created corpus. The results have shown that a Bidirectional Encoder Representation from Transformers (BERT) based model excels with 78% accuracy in polarity detection, while deep learning and Random Forest models reach around 70%. For emotion detection, the BERT model performs best with 68%, followed by deep learning with 55%. It is worth noting that emotion detection is more challenging due to the subjective interpretation of emotions in the complex communicative context of video gaming on platforms such as Twitch. The use of supervised learning techniques, together with the rigorous corpus labeling process and the subsequent corpus pre-processing methodology, has helped to mitigate these challenges, and the algorithms have performed well. The main limitations of the research involve category and video game representation balance. Finally, it is important to stress that the integration of machine learning in video games and on Twitch is innovative, by allowing the identification of viewers' emotions on streamers' channels. This innovation could bring benefits such as a better understanding of audience sentiment, improving content and audience retention, providing personalized recommendations and detecting toxic behavior in chats.

[*] Corresponding author.
E-mail addresses: noemer@tel.uva.es (N. Merayo), rosalia.cotelo@uam.es (R. Cotelo), rcarrata@uji.es (R. Carratalá-Sáez), fandujarm@infor.uva.es (F.J. Andújar).

https://doi.org/10.1016/j.csl.2024.101651
Received 21 December 2023; Received in revised form 6 March 2024; Accepted 20 April 2024
Available online 25 April 2024

List of abbreviations:

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Networks
IG	Information Gain
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
SVM	Support Vector Machine
RF	Random Forest
SA	Sentiment Analysis

1. Introduction

In October 1958, the Physicist William Higinbotham created "Tennis for Two", which is considered the first ever video game (Tretkoff, 2008). With it, the video game industry was born, always focused on the business of providing entertainment. By the end of the 20th century, the success of network games became consolidated in such a way that authors like Lafrance (2003) baptized them as the first mass medium of the electronic era and the most important of the new generation of interactive media. The video games industry has reached a worldwide volume of around one hundred million euros, with 1.1 billion online gamers worldwide (Clement, 2023), even overcoming the economic crisis that has affected the rest of the leisure industries (Gómez García et al., 2007; Vera, 2015).

Ever since the proliferation of online multiplayer games, this form of entertainment has acquired a social dimension that is closely related to the defining aspects of the participatory culture and virtual communities (Jenkins, 2009; Velez et al., 2018). Bound to online multiplayer games, many platforms have appeared as environments to develop culture and virtual communities around them by sharing gameplays, opinions, and live gaming sessions. Since its launch in June 2011, Twitch has quickly become the platform of choice for video game streamers, especially among the younger generation, surpassing competitors such as YouTube Gaming or Facebook Gaming. Twitch is an open platform where multimodal communication takes place. The streamers create content by sharing their computer screen while playing video games, and the viewers watch the streamers' content live as they play. In contrast to traditional media like television shows or programs, in Twitch the users only chat during the time of the stream (the live broadcast), allowing for live streamer-viewer communication, thus impacting the stream.

Analyzing Twitch from a language perspective, Spanish is currently the second largest language regarding number of users and broadcasting channels (TwitchTracker, 2024). Reinforcing this is the fact that at the beginning of October 2023 (TwitchTracker, 2024), there was one Spanish channel in the top 10 most viewed channels, four Spanish channels in the top 20, and 15 in the top 100 (Twitch Tracker, 2023). Considering the traffic of the different platforms concerning the world of video games, Twitch is the most visited webpage in the "Video games" segment in Spain, with a 10% share (Similarweb LTD, 2023). Its success makes Twitch a unique source for research, both because of the number of users who produce and comment on content of a very diverse nature, and because of the richness and peculiarity of the type of multimodal communication that takes place on it. Consequently, several authors have reflected on the Twitch communities (Cai et al., 2021; Seering and Kairam, 2022), the motivation and behavior of Twitch users (Gros et al., 2017), the impact on the users' well-being (Zsila et al., 2023), and its security concerns (Cai et al., 2023), among other topics.

The large and growing impact of video games on social platforms such as Twitch, coupled with the large amount of information and opinions poured into these channels in real-time, makes the application of Artificial Intelligence (AI) of great relevance. AI techniques are necessary to automate, analyze, and facilitate the immediate monitoring and categorization of such information. Specifically, Sentiment Analysis (SA) is a branch of AI and Natural Language Processing (NLP) is concerned with the identification, extraction, and comprehension of emotional reactions in written or spoken text. Several approaches to applying sentiment analysis in texts commonly include lexicon-based techniques and supervised learning. Sentiment lexicons use dictionaries that associate words with sentiment labels (positive, negative, neutral) or emotions. Typically, the scores of the words in a text are added together to calculate an overall score for the emotional response. On the other hand, supervised learning trains a machine learning model using a labeled dataset that includes examples of text with their respective sentiments. This way the model learns to classify new texts into sentiment categories. Lexicon-based techniques can have significant limitations, as many lexicons focus only on positive and negative words leaving out neutral words, and are often designed for a specific context. Moreover, the number of lexicons in Spanish is rather limited (Redondo et al., 2007; Moreno-Ortiz and Hernández, 2013). It is worth noting that the nature of comments posted on platforms such as Twitch shows strong linguistic challenges (Cotelo García, 2022) due to the brevity of messages, lack of context, grammatical errors, use of irony and sarcasm, informal language style, multilingualism, or excessive use of emoticons/emotes, among others. These characteristics greatly complicate sentiment analysis in these contexts.

There is little research in the literature on the application of AI techniques in the evaluation of emotional response (polarity or emotions) in the field of video games and within virtual contexts, particularly in online platforms such as Twitch. Specifically in the case of Spanish video game streaming sessions, we have not been able to find any existing work that focuses on using AI to identify emotions and polarity in Twitch chat messages. This context provides an exceptional research environment due to the strong social and economic impact of the interaction between video games and platforms followed by millions of users worldwide, which allows us to understand the emotional responses generated in these digital environments.

Taking all the above into account, we formulated the following hypotheses:

- AI techniques that leverage machine learning can serve to classify the emotions and polarity of the chat messages of Twitch video game streams.
- It can be possible to measure the emotional impact of Twitch streaming sessions on the viewers by analyzing their chat messages, even determining the emotions that prevail in them.

In this work, we focus on providing tools for SA of Twitch chat messages, motivated by our hypotheses and the mentioned interesting outcomes that can derive from it. This research is therefore pioneering in different areas. On one hand, this is the first research, both in Spain and internationally, focused on analyzing the emotional response to video games streamed on Twitch. On the other hand, the integration of AI is exceptionally pioneering in the field of video games and Twitch, making it possible to automatically identify the emotional response generated in streamers' channels to video games, game strategies, or even about the streamers or players themselves.

Thus, the main research objectives and contributions of our work are:

- **Novel corpus**: Designing a novel corpus labeled with polarity (positive, negative, neutral) and emotions from Spanish Twitch channels in the video games environment. This corpus is available on GitHub (Merayo et al., 2023) so other researchers could use it in their artificial intelligence applications or any other related application context.
- **Machine learning algorithms**: Modeling a set of machine learning algorithms to detect emotional response, polarity, and emotions in Spanish Twitch channels broadcasting video game content.

This paper is organized as follows. Section 2 shows the state of the art in video games, artificial intelligence, and Twitch. Section 3 contains the process of building the novel corpus with chat messages of video game channels on Twitch. Section 4 shows the main statistics of the corpus. Section 5 describes the classification models used to predict the emotional response (polarity and emotions). Section 6 shows the results of the classification models and the discussion of the results. Finally, Section 7 summarizes the main findings and conclusions of the conducted research.

2. Related work

Given the sheer number of Twitch users (including streamers and viewers) and the vast number of live video game broadcasts taking place simultaneously on the platform, analyzing the content of chat messages is humanly intractable. Even if only one stream is considered, thousands of messages are launched in the chat during a live session. For this reason, the use of computational resources is key when it comes to analyzing everything related to communication through Twitch chats and, in particular, the interdisciplinary NLP field, which combines computer science and linguistics to extract useful information from texts.

Regarding the usage of NLP for processing texts of video game users, in Agca (2023), the authors present an emotional analysis of the words that form the game summaries (taken from different websites dedicated to video game rating), using NRCLex and Natural Language Toolkit (NLTK) to look for relationships between the video games' user scores and summary sentiments. NRClex measures the emotions of a text using the 2016 National Research Council Canada lexicon dictionary. NLTK is a platform for building Python programs to work with human language data. In Guzsvinecz and Szűcs (2023), the authors use NRC Emotion Lexicon from the statistical program package R to automatically analyze video game reviews to evaluate and relate length, positiveness, game duration, and emotions. In contrast to our work, none of these papers apply machine learning from specialized corpora, but use existing dictionaries and tools. Moreover, the objective is not the same as ours since those works evaluate user summaries and reviews, while we focus on identifying emotions within Twitch live chats. On the other hand, the authors in Barbieri et al. (2017) model Twitch emotes usage to identify *trolling* attitudes by comparing the performance of three different models: A bidirectional Long Short-Term Memory recurrent neural network (LSTM) model, a Bag-of-Words model, and a Skip-gram model, concluding that the first model, based on the Recurrent Neural Network, is the most accurate approach. The procedures explored in this paper are closer to ours, but the authors focus on a completely different element: Emotes instead of text. Furthermore, this research focuses on detecting trolling attitudes, but not on analyzing the emotional response generated in Twitch chats.

With the purpose of analyzing the video game content or user impressions, some authors propose leveraging video games to generate a corpus for sentiment analysis that could later be used in other contexts. In Pöyhönen et al. (2022), the authors claim that video games offer a considerable source of dialogues and texts that could be used to generate databases with which to later nourish and train NLP tasks. Concretely, the authors leverage the BERT model to automatically detect persuasive intentions in the texts. The same principle is defended in Hämäläinen et al. (2022), where the authors make use of already emotionally labeled dialogues in the Fallout New Vegas video game (by the game developers) to generate a corpus for emotion identifications using multilingual BERT, XLMRoBERTa, and language-specific BERT models. The idea of using NLP models such as BERT resembles our approach, but no new specialized corpus is provided in this work. Moreover, the evaluation employs video game dialogues instead of user chat messages, thus the authors can rely on regular dictionaries, as the language is much more standard than that observed in Twitch chats. Furthermore, the objective of the mentioned work is to build data sets to train AI tools that can later serve other fields, while our objective is to evaluate the emotions observable in the chat during Twitch streaming sessions.

The design of a specialized corpus for the gender bias analysis in video games is what the authors propose in Stephanie et al. (2023), where a large-scale corpus of video game dialogue is presented to measure and monitor gender representation in video game dialogue. The creation of a specialized corpus is similar to our work methodology. Nevertheless, the mentioned work focuses on identifying gender biases to provide a useful tool to detect gender inequalities in video games and contribute to their correction, while we intend to provide a corpus that can serve to analyze the emotional response in live interactions on Twitch.

There are also research works that analyze the content of video games using non-AI tools. For example, in Olejniczak (2015), the author analyzes the length, uptime, and lexical particularities of the language used in several Twitch chat messages using WordSmith Tools 6.0 and applying Wordlist and Keyness. The findings of this study are that messages tend to be short, with a low uptime, and introduce novel lexical items and distinctive emoticons that influence the general language of the players. Alternatively, the authors in Deng et al. (2015) use the Linguistic Inquiry and Word Count (LIWC) tool and existing emotion-related dictionaries to assess the impact of COVID-19 on emotions in chat messages, concluding that Twitch users felt more anger and anxiety and employed fewer social words. In Casado and Peña-Acuña (2022) the authors discuss the Spanish vocabulary used in video game streaming chats, using existing lexicons to conclude that there is a combination of anglicisms, neologisms, and Spanish terms. Compared to our research, these three works share the analysis platform (Twitch) and the text types (chat messages), but they do not propose any specialized video game corpus, nor do they explore the use of NLP models.

From the conclusions derived from the different works referenced, it is clear that video games have captured the attention of many researchers due to their impact on today's society. However, very little research has focused on analyzing emotional responses in the context of video games on virtual platforms such as Twitch. Although NLP has been integrated into different application domains with great success, to the best of our knowledge, there are no works that offer corpora designed for sentiment analysis in video games to extract the emotional response in Twitch chats, and there is no research applying sentiment analysis in Spanish Twitch channels. Applying sentiment analysis to video game channels on Twitch has a significant impact in an environment that deals with a large amount of people and money. This technology, applied to Twitch streaming, can provide valuable insight into viewer sentiment, helping streamers understand the audience's opinions about games, broadcasts, and several aspects of their streaming style and content. This knowledge will facilitate content and strategy decisions, improving viewer retention by tailoring content to their emotions and preferences. Moreover, it can also assist in the detection of toxicity and inappropriate behavior, ensuring real-time compliance with ethical standards in Twitch chats.

3. Creating the corpus

In this section, we will outline the methodology used to determine the type of Twitch streamers selected, as well as the process for selecting posts and messages. Additionally, we will cover the process of annotating the corpus and labeling it with its emotional response, including polarity and emotion.

3.1. Selection of Twitch streamers

Our corpus includes chats from different Twitch streams, selected based on specific criteria:

- *Channels in Spanish*. We selected local streamers, who belong to the Spanish Twitch community. Sharing the native language of these streamers facilitates the interpretation of certain cultural and communicative keys specific to this linguistic community. It is also worth noting that Spanish is the second most spoken language on the Twitch platform, being only surpassed by English (TwitchTracker, 2024). For our corpus we ruled out bilingual streamers, whose chats could alternate between English and Spanish. Even so, we are aware of the high proportion of English terms registered in Twitch chats, both because it is a digital network mainly used by young people and because it contains a specialized lexicon in video games (Casado and Peña-Acuña, 2022).
- *Channels whose main content is video games*. All downloaded chats belong to Twitch broadcasts in which the streamer was playing a video game or commenting on a video game event. The reason is that one of the objectives of our project is to study the type of messages produced in a semi-specialized field of communication, such as the world of video games. This area also represents the most characteristic content of the Twitch platform, so studying it also reveals defining features of this platform. We have gathered chats corresponding to very different types of video games: World of Warcraft, League of Legends, Fortnite, Apex Legends, Diablo 4, Red Dead Redemption 2, Dark Souls, and Heroes of the Storm.
- *Channels with a viewer limit of no more than 1000 people*. Recent literature has established that in chats with a higher number of participants, communication or interaction becomes almost impossible among those participating in the chat: The rate of incoming messages causes the so-called "scroll factor" (Yus, 2020), and it tends to become "an illegible waterfall of text" (Hamilton et al., 2014). Based on the definition by Ford et al. (2017), who categorize small chats on Twitch as those with less than 2000 viewers, we have decided to set a threshold of 1000 viewer streams for our study. This ensures that the chats of these streams meet the specific criteria that Ford et al. establish for "small chats". Small chats should present longer messages, increased original content, and a greater number of unique voices (i.e. perspectives or stances), all of which promote the richness and clarity of the emotions for our corpus.
- *Channels in which the streamer has a webcam*. Preference is given to these streams because engagement is usually higher on channels where the streamer shows their face via a webcam integrated into the live broadcast (Jodén and Strandell, 2022).

Table 1
Video game genre and type of player interaction of video game selected to create the corpus.

Game	Main video game genre	Type of player interaction
World of Warcraft	MMORPG	Cooperation with team members
League of Legends Heroes of the Storm Fornite Apex Legends	MOBA MOBA Battle royale shooter Battle royale shooter	Competitive games based on teams (cooperate with team, compete with other teams)
Diablo 4	Action RPG	Single Player Game
Red Dead Redemption 2	Action/adventure	Single Player Game
Dark Souls	Action RPG	Single Player Game with cooperative and competitive mechanics in online-game

For the process of creating the corpus, we started from an initial selection of six streams covering Blizzard's event of an update for the World of Warcraft video game that took place on April 19, 2022. Choosing events that can generate a particular variety of emotions and polarized messages has previously been used for similar studies, such as in analyzing messages that occur around "great relevant events in a specific time frame on Twitter" (Plaza del Arco et al., 2020). With the chats extracted from these streams, the first tests were carried out by manually annotating the messages.

Subsequently, the corpus has been expanded in successive layers, always with streams that meet the previously described criteria and that focus on comments on gaming events, or streams depicting gameplay. We varied the types of streamed games, including competitive, cooperative, and single-player games, and covered different video game genres, as illustrated in Table 1. Additionally, we aimed to have a similar number of male and female streamers to ensure a balanced representation. We also noticed that the chat dynamics tend to be slightly different between male and female streamers, with more affective responses observed in the chats of female streamers. After following this methodological process, the complete list of the channels included in our corpus is: Anytimeshield, belvid, bollostream, evangelion0, hawnktv, ipandarina, kryp, lovilu, lulypop, nano_hots, odii, paracetamor, paxelol, phoebina, raevencca, sevenjungle, tigry86, xixauxas, yyyugo_tv, and zullhammer.

Two free software tools were used to download and extract videos and chats from the selected streams:

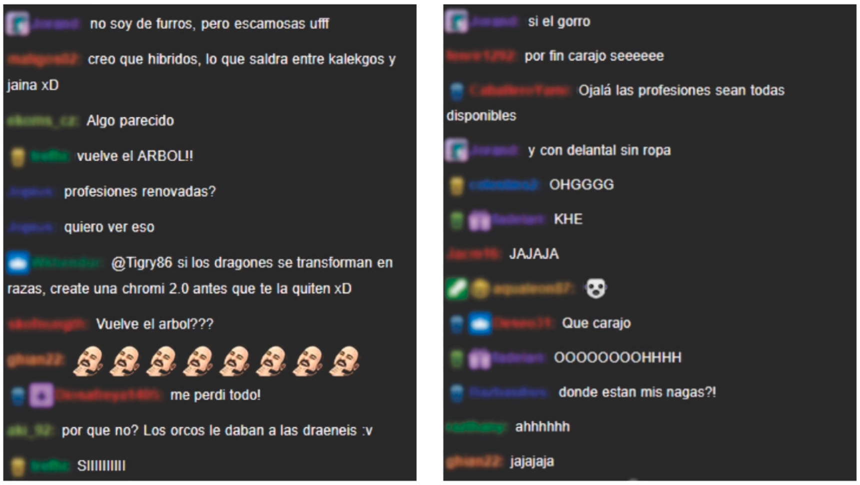
- **Twitch Leecher** (Rebitzer, 2022), which is a video download manager that can provide users instant access to downloading Video On Demand (VOD) files stored on Twitch. This application is especially useful because each VOD can be selected to be cropped to any desired start and stop spot. Given that some Twitch streams lasted for several hours, we were interested in downloading the videos partially.
- **TwitchDownloader** (Pardo, 2023), which is a software that allows users to download chat from VODs and Clips in either a JSON (JavaScript Object Notation) with all the original information, a browser HTML (HyperText Markup Language) file, or a plain text file. It is also possible to update the contents of a previously generated JSON chat file with an option to save as another format or to use a previously generated JSON chat file to render the chat with different static and animated emotes used on Twitch.

3.2. Processing and selection of Twitch messages and corpus annotation process

Per the Twitch Terms of Service (Twitch.tv, 2023), users need to verify their age before accessing the platform. Individuals below the age of 13 are not allowed to use Twitch. However, it is virtually impossible to confirm the age, gender, origin, or profile of each user, as registration on the platform is free and does not require any personal data. Following the guidelines provided by Heise et al. (2020), we can consider that the streams and messages from the viewers in the chat should be treated as public places. However, to maintain the highest level of anonymity and comply with those guidelines, we have anonymized the usernames of all participants in the chats included in our corpus. We also filtered out messages from Twitch bots (Streamlabs (Logitech Services S.A., 2014) and Nightbot (NightDev, LLC., 2023)), @ mentions, and URLs (this process will be explained in detail in Section 6.1). These previous considerations enable us to incorporate ethical and privacy principles into the process of creating our corpus. This will permit us to guarantee that the collection and use of data are done fairly, respecting privacy, avoiding bias and discrimination, and ensuring transparency in the process.

Additionally, we removed messages that only consisted of emotes without any accompanying text. Emote-only messages are commonly used on Twitch to express a shared emotional response that is echoed throughout the chat. Therefore, they are more useful for identifying emotional patterns and trends across entire chat conversations rather than for detecting specific emotions in individual messages. However, messages composed only of emotes were kept if those emotes were of general and frequent use on Twitch (Ravenbtw, 2023) and if they were univocal emotes in terms of polarity and emotion. For this type of emotes, a list was elaborated in which 80 emotes from our corpus were selected. Typical examples of these emotes are BatChest (Approval), BibleThump (Sadness), KEKW (Approval), or Madge (Disapproval); the complete list can be found in the corpus.

During the corpus annotation process, each pair of annotators worked together to compare and contrast their labeling of both polarities and emotions. To ensure the accuracy and consistency of the manual labeling process, each annotator was provided with



[redacted] no soy de furros, pero escamosas uff

[redacted] creo que hibridos, lo que saldra entre kalekgos y jaina xD

[redacted] Algo parecido

[redacted] vuelve el ARBOL!!

[redacted] profesiones renovadas?

[redacted] quiero ver eso

[redacted] @Tigry86 si los dragones se transforman en razas, create una chromi 2.0 antes que te la quiten xD

[redacted] Vuelve el arbol???

[redacted]

[redacted] me perdi todo!

[redacted] por que no? Los orcos le daban a las draeneis :v

[redacted] SIIIIIIIII

[redacted] si el gorro

[redacted] por fin carajo seeeeee

[redacted] Ojalá las profesiones sean todas disponibles

[redacted] y con delantal sin ropa

[redacted] OHGGGG

[redacted] KHE

[redacted] JAJAJA

[redacted]

[redacted] Que carajo

[redacted] OOOOOOOOHHHH

[redacted] donde estan mis nagas?!

[redacted] ahhhhhh

[redacted] jajajaja

Fig. 1. Screenshots of chats in a World of Warcraft stream during a gaming event.

an annotation guide that contained detailed guidelines on the meaning of each emotion category. The guide also included real examples extracted from the corpus that helped clarify the scope of each label (Merayo et al., 2023). Two independent experts were assigned to label the corpus separately. After the initial labeling phase, a third expert reviewed the results to resolve any discrepancies. If a disagreement among the three experts arose, they discussed and deliberated on the matter until a consensus was reached. When a consensus could not be reached, the message was discarded. This approach was necessary to prevent any potential errors or biases in the final labeled corpus. The experts who participated in the categorization process showed a strong inter-rate reliability (Hallgren, 2012), with a discrepancy of only 5.25% in the polarity labeling and 8.45% in the emotion labeling. These results provide further confidence in the consistency of the labeling process that was conducted.

In the annotation process of the corpus, we noticed some chat messages that were too fragmented, lacked enough context, or had little linguistic content. As a result, we had to discard them from the final set of messages. These types of messages are frequent in Twitch chats, as the platform promotes, by its very nature, discursive fragmentariness and hyper-specific contextualization. Examples of fragmentariness in chat are shown in the screenshots in Fig. 1. Some users split their contribution into brief messages: “profesiones renovadas? / quiero ver eso” (“renewed professions? / I wanna see that”), as others chain messages that contain only expressive and onomatopoeic utterances, such as “ohggg”, “oh” and “ahhhh”, expressing surprise or emotion, and “jajaja”, expressing laughter.

Furthermore, in Twitch, both the speed at which the stream and the chat progress propitiates a communicative urgency that translates as morphological abbreviations and evident fragmentariness in the syntactic construction of messages, making it difficult to automatically interpret them. On the other hand, in most cases, chat messages are commenting on events that are happening in the stream itself, so the referentiality is diluted, and the understanding of the messages is impossible in some cases since messages are “highly context dependent” (Recktenwald, 2017). Additionally, in the context of video game streams, a remarkable degree of linguistic specialization and jargon exists, which can also make it difficult to understand the messages for users unfamiliar with that vocabulary. For this reason, the messages of the corpus were selected according to the quantity and quality of their linguistic content and the possibility of being interpreted without the need to know the exact context in which they are produced. Messages with jargon were maintained, since the semantic stability inherent to the level of specificity of a voice of a terminological nature guarantees that the meaning of the word rarely changes, regardless of context.

3.3. Description of the corpus labels: Polarity and emotions

As previously indicated, one of the objectives of this study is to classify the messages of the corpus in terms of their polarity and emotional category. This implies a previous process in which we had to choose which labels to select, depending on the communicative characteristics of Twitch and the examples observed in the initial sample of chats that were going to be manually labeled.

As for polarity, we adjusted it to the binary distribution usually considered, establishing the labels of Positive (P) and Negative (N), and adding a Neutral (NEU) label only for those cases in which it is not possible to label the message with any polarity.

As expected, the determination of the labels of emotions was more complex. We examined several well-known models, including Ekman’s (Ekman, 1999) and Plutchik’s (Plutchik, 1980), but eventually decided to adopt the Plutchik model. The main reason for this choice is that Ekman’s model is based on six primary emotions, i.e., fear, anger, sadness, disgust, happiness, and surprise, whereas Plutchik’s model is based on a wheel that includes eight basic emotions, i.e., joy, confidence, fear, surprise, sadness, disgust, anger, and anticipation. These eight basic emotions include trust and anticipation, which are essential for identifying some of the most representative emotions that can be observed in Twitch chats. Trust is related to emotions such as confidence, admiration, and acceptance, which are expected to be prevalent in chats primarily composed of followers and viewers who enjoy the stream’s content. The category of anticipation aligns perfectly with the concept of hype, which is critical, as we will elaborate later, in the context of the Twitch platform. From our Twitch knowledge base and the initial corpus that we collected, we anticipated that the messages' brevity and fragmentation would make the categorization according to a wide range of emotions difficult. Therefore, we selected a limited number of well-defined categories exclusive to the video game research environment and Twitch. These categories needed to encompass the usual range of emotions observed in the gaming environment on Twitch, which would coincide with those experienced by the audience of a show or sporting event (Sjöblom and Hamari, 2017).

The final set of labels includes the following emotions: Approval (Approval/ Confidence/Empathy), Disapproval, Sadness (Disappointment/Sadness), Anger, and Hype (Acceptance/Hype/ Interest). To these, we added the "Neutral" category for cases where it is impossible to categorize the message with an emotion. A more exhaustive description of each emotion is set out below.

- **Approval** (Approval/Empathy/Confidence): This label applies to messages that express agreement and approval, usually directed to what is happening in the stream, to the topic treated in the stream, or that demonstrate gestures of empathy, understanding, solidarity, engagement or trust with the streamer, regarding what the streamer says or does. Messages in this emotional category do not always have to be positive, as trust or empathy can be expressed in negative situations where the speaker shows understanding and support; for example: "ah no joder q unlucky" ("ah that sucks, that's so unlucky").
- **Disapproval**: The counterpart of the previous label is "disapproval", which describes messages that show displeasure and disagreement of the speaker in a moderate way (if the expression of disapproval is very intense, we will find those messages under the label Anger). Messages where chat users indicate that the content or topic bores them, is repetitive or does not interest them for personal preferences, are also included under this category.
- **Sadness** (Disappointment/Sadness): We will speak only of "disappointment" when referring to expectations that have not been fulfilled, and therefore, contrast between what was expected to happen and what has happened is established. This category also includes feelings of sadness that are expressed explicitly, either through words (sad, sadness, cry, downfall, etc.) or emoticons.
- **Anger**: When an emotion of a negative nature (disapproval, disappointment, etc.) is expressed intensely and categorically, or when messages of resentment and anger containing rebukes, insults, etc., are present in the chat, we will use this label. For example: "a callar pendejos" ("shut up you assholes").
- **Hype** (Interest/Acceptance/Hype): A label that refers to the expression of the feeling caused by the desire and expectations for a future event: A game or the update of a game of upcoming release, an e-sports match that will start soon, etc. For example: "siento esa emoción como si soy un niño pequeño esperando la navidad" ("I feel that emotion as if I am a little child waiting for Christmas"). It is a label that seems restrictive because it only applies to messages expressing an emotion of illusion, anticipation, interest, etc., projected into the future, or where the word "hype" is explicitly mentioned. Still, it is nevertheless one of the most characteristic emotions of the Twitch platform. It is such a structural emotion on the platform that there is a tool implemented in its interface to reinforce that emotion, called the "hype train", a chat event that generates rewards for viewers and streamers.
- **Neutral**: Some of the messages lack sufficient context to be conveniently labeled, as Twitch messages occur in chats in a very fragmented way, in an agile and conversational dynamic that runs, sometimes, at high speed, closely linked to what is happening on the screen. Additionally, in digital communication, we must consider the use of humor, sarcasm, and hyperbole, which can also make categorization difficult. For instance, we can consider the message: "adiós a mi vida social" ("goodbye to my social life"). In the context of video games, renouncing social life generally implies that a game is greatly liked or engrossing. Therefore, the message should be interpreted with a positive connotation. This viewer is not expressing sadness about losing their social life, but rather using it humorously to indicate their anticipation of spending many hours enjoying a particular game.

In addition to these general guidelines, lists of recurring lexical elements were also created so that they could serve as a guide for labeling in difficult cases, as we were aware of the possible overlap between emotions and the ambiguity of some of the messages found on Twitch. For example, for the label Hype we consider messages that contain the term "hype" itself, expressions like "it's coming", "looking forward to"... or references to physical reactions (goosebumps, chills, tremors, tears, etc.). This helps distinguish the Hype emotion from the Approval emotion, which aligns with expressions of conformity, positive adjectives ("cool", "nice", "fantastic", etc.), and phrases such as "I like", "I love", etc. We can observe these differences in two messages from our corpus that are expressing a reaction to a rework of certain characters:

(a) "el rework de hecarim esta guay" ("hecarim's rework is cool")
(b) "class reworks se vienen" ("class reworks... it's coming")

Both are labeled as Positive Polarity, but while (a) is categorized as Approval ("is cool"), (b) is labeled as Hype ("it's coming"). Similarly, in a stream from our corpus in which viewers are reacting to the trailer of a new game, messages of different characteristics can be found:

(a) "muy buen trailer" ("very good trailer")
(b) "me estan dando ganas de llorar, que bonito es el trailer" ("it's making me want to cry, how beautiful the trailer is")
(c) "ahora se viene el trailer de verdad" ("the real trailer is coming now for real")

Table 2
Distribution of the corpus in the different polarity categories (P, N, NEU).

Polarity class	Number of samples	% of each class
P	1000	45.1%
N	968	43.7%
NEU	247	11.2%

Table 3
Distribution of the corpus in the different emotion categories.

Emotion class	Number of samples	% of each class
Approval	711	32%
Hype	268	12.1%
Disapproval	482	21.7%
Sadness	271	12.2%
Anger	168	7.6%
Neutral	315	14.2%

We labeled (a) as Approval, since the message is simply expressing that the trailer is good, but (b) and (c) would be examples of Hype, due to the expression of intense, visceral emotion (the desire to cry) and the anticipation expressed in "is coming".

On the other hand, a Neutral expression is one in which the emotion is not sufficiently marked to be specified in a label, compared to, for example, Disapproval messages that indicate clear feelings of rejection. This can be illustrated with some examples from our corpus, that were posted during a stream featuring the announcement of updates for the World of Warcraft game. During the stream, discussions about the inclusion of a game mechanic known as "housing" or the addition of dragons to the game generated chat messages like the ones below:

(a) "no hay housing" ("there is no housing")
(b) "no hay housing me mato" ("there is no housing I kill myself")
(c) "nooooooo" ("nooooooo")
(d) "noooo no quiero dragones" ("noooo I don't want dragons")

While (a) is a merely descriptive message, which is therefore labeled with Polarity and Neutral Emotion, (b) expresses an obvious rejection of the idea commented on the stream. On other occasions, the Neutral label is determined by insufficient context, as in the case of (c), which is impossible to interpret without knowing the context in which it occurs (it could be Disapproval, Sadness, or even Approval, through Empathy). Instead, (d) adds enough linguistic content to the "noooo" to determine an adverse reaction to what was happening on the stream, which allows the message to be labeled with the Disapproval emotion.

4. Description of the video game corpus in Twitch

Our corpus is provided in XLSX format, which can easily be converted to CSV format if needed. It is made up of 2216 samples (Twitch messages) and consists of three variables, Text, Polarity, and Emotions, which are described as follows:

- **Text**: It is an independent variable, representing the input feature. These are the raw comments from Twitch users reacting to the stream. This variable will require further processing to be analyzed. Details of this processing will be provided below in Section 6.1.
- **Polarity**: It is one of the target variables that can take three values depending on the polarity of the message: Positive (P), Negative (N), Neutral (NEU).
- **Emotion**: It is one of the target variables that can take six values depending on the emotion of the message: Approval, Sadness, Disapproval, Anger, Hype, Neutral.

The independent variable (Text) represents the raw comments, while the dependent variables (Polarity and Emotions) are used to classify and label the comments according to their associated polarity and emotion. Regarding the dependent variable Polarity, Table 2 shows that the 2215 samples are mainly distributed between the Negative (N) and Positive (P) polarities. The distribution of the corpus in classes is 1000 P comments, 968 N comments, and 247 NEU comments. As for the dependent variable of "Emotion", it assigns the majority of the samples to the class Approval, accounting for around 32% of all samples. Specifically, the distribution of the corpus in classes is 711 comments for Approval, 482 comments for Disapproval, 271 for Sadness, 268 for Hype, 168 for Anger, and 315 for Neutral (Table 3).

Furthermore, our particular case is a multiclass classification problem as it shows several possible polarities and emotions. Since each Twitch message can only belong to one polarity/emotion, it is known as a single-label multiclass classification problem. One issue to take into consideration in labeled corpora is the balance between classes or categories. In classification tasks, an unbalanced distribution, where one class significantly outperforms another, can cause classification models to favor the prediction of instances of the majority class. This bias has a detrimental effect on the performance of algorithms.

Finally, the video game corpus is available on GitHub under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA) license, to allow researchers to use it in their applications in both artificial intelligence and other application contexts (Merayo et al., 2023).

5. Classification models

In this section, we will describe the different machine learning models that will be used in the detection of the emotional response of our corpus. SVM and RF were selected for their low complexity and computational efficiency compared to more complex models like deep learning models based on neural networks. RF, in particular, performs well without hyperparameter tuning, significantly mitigates overfitting risks, and maintains stability with new samples since by using hundreds of trees the average of their votes still prevails. On the other hand, the advantages of Deep Learning models over simpler models (SVM, RF) include the ability to learn more complex and abstract feature representations, automatically adapt to patterns in large, non-linear data, and improve their performance with larger datasets. Finally, BERT models, despite their higher computational complexity and resource requirements compared to simpler models, offer significant advantages and superior performance. Specifically, BERT enables the capture of general language knowledge as it is pre-trained on large datasets, aiding in the improved handling of ambiguity and polysemy. Furthermore, by processing context bidirectionally, it captures complex relations, dependencies, and connections between words in a sentence, facilitating the learning of richer and more complex semantic representations.

5.1. Support Vector Machine

Support Vector Machine (SVM) is a supervised learning technique used to classify data. It searches for an optimal plane that can separate different classes, in our case polarity levels or emotions. This technique can handle high-dimensionality datasets and use kernel functions to find more complex separation boundaries (Noble, 2006). The hyperparameters to be optimized in this model are as follows: Regularization parameter, Kernel type, and Kernel parameters.

5.2. Random Forest

Random Forest (RF) is a machine learning algorithm based on combinations of decision trees. Thus, each decision tree is built using random samples of the training data to improve the quality of the predictions. Furthermore, RF takes random subsets of features to find the best split. Once all the trees have been built, each tree outputs its own prediction. The final prediction is determined by voting, and the option that gets the most votes from the trees is chosen (Probst et al., 2019).

5.3. Deep learning model

Our deep learning model integrates the capabilities offered by both convolutional and recurrent layers to harness the strengths of each model. Convolutional Neural Networks (CNN) have powerful features when applied to NLP, as they can learn distributed word representations or capture sequential patterns, making them a solid choice in many applications (Chollet, 2021). However, CNNs have no memory and are not able to capture long-term dependency relationships in text sequences. In contrast, Recurrent Neural Networks (RRN) have memory, i.e., they store information from the texts already processed, which can be very useful for analyzing future texts. The use of a LSTM layer enhances this property of recurrent neural networks by incorporating long-term memory along with short-term memory (Chollet, 2021). The proposed hybrid model is composed of the layers shown in Fig. 2. This model will classify the Twitch comments into three polarity levels (P, N, NEU) or five emotions (plus a NEU category). The convolutional network will therefore transform the tensors representing the text (Twitch comment) into a sequence of more representative high-level features that the recurrent network will receive as input. It is important to remark that we use the categorical cross-entropy loss function and the Adam optimizer.

This is a summary of the functionalities and utilities of each layer:

- Embedding layer: Transforms messages (Twitch comments) into tensors with which the neural network is fed by applying the word embedding technique, which represents each word as an n-dimensional vector to use similar values to represent words that have a semantic relationship. The result is a matrix where the rows represent each of the tokenized words that make up the messages, and the columns represent the weights assigned to each word. These weights are adjusted using the backpropagation algorithm as the network is trained, starting with random word vectors.
- Convolutional layer: Reduces the processing that the next layer has to do suppressing intermediate steps by detecting text patterns. We use the ReLU activation function due to its efficiency, which introduces non-linearity and helps CNNs learn complex features from the input data.
- MaxPooling1D layer: Reduces the dimensionality of intermediate representations and extracts relevant features from input sequences to have an RNN layer afterward.
- LSTM layer: Improves upon the characteristics of recurrent neural networks by integrating both long-term and short-term memory. We adjust the number of neurons, the dropout rate, and the recurrent dropout rate. The last two parameters aim to reduce over-fitting, by deactivating neurons for normal connections (dropout) or deactivating neurons for recurrent connections (recurrent_dropout).

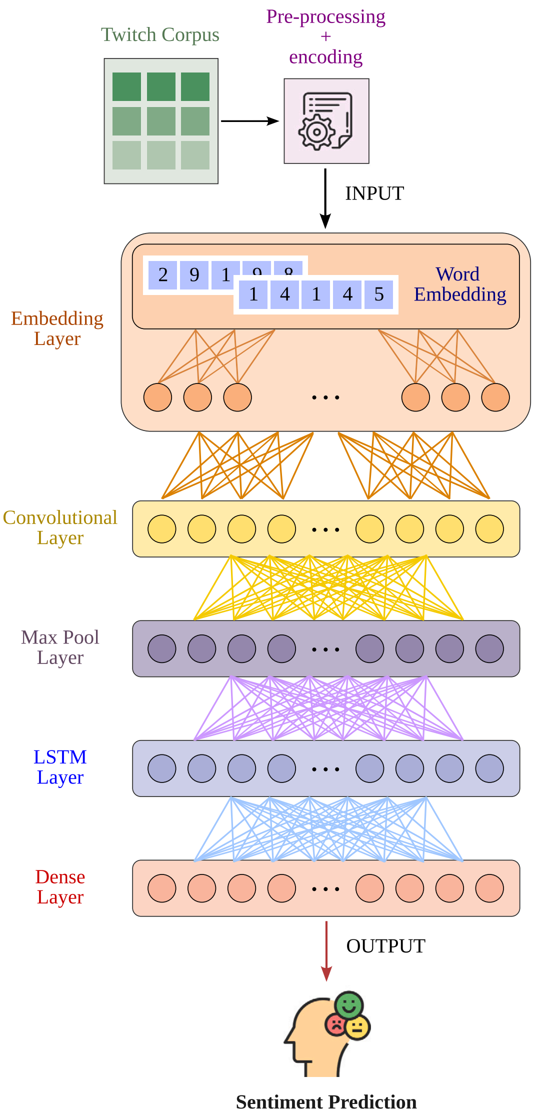


Fig. 2. Block diagram of our proposed hybrid deep learning model with the different layers.

- Dense layer: Generates an output representing the probability of belonging to each possible class using the Softmax activation function. In our case, the classification considers polarity (P, N, NEU) and emotions (five emotions plus a NEU category), so this layer will have three or six neurons depending on the classification problem.

Concerning the loading of the embeddings in the word processing task, we acquired them through training the network with data extracted from the corpus itself. Pre-trained dictionaries are more generic and their effectiveness heavily relies on the alignment between the words in our corpus and those in the dictionary. For this reason, the vocabulary is created by counting the occurrences of each word in the corpus, using the information gain technique (IG) in order to keep the words with the highest occurrence. We opted for this approach instead of using absolute frequency because the most common words are often not the most effective in distinguishing between categories. This is because a word can appear an equal number of times in different classes, which means that it would not provide relevant information. In contrast, the IG of a word reports the frequency of a word in a category (C) compared to the frequency of the same word in the other categories. To calculate the IG of a word, we first calculate the entropy (Witten et al., 2011; Larose and Larose, 2014) with Eq. (1):

$$H = \sum_{i=1}^{n} \left(p_i \times log_2(p_i)\right) \tag{1}$$

where p_i is the probability $P(w_i \parallel c_i)$ of the word w_i appears in the class c_i. Then, the IG follows Eq. (2):

$$IG(C,X) = H(C) - H(C,X) \tag{2}$$

where C is the class set and X is the subset of texts (Twitch messages) in which the term w_i appears. $H(C)$ represents the probabilities of each category within the corpus, and $H(C,X)$ the probabilities of a word appearing or not in the corpus and in each of the categories.

5.4. *Bidirectional encoder representations from transformers models: RoBERTuito*

BERT models use an architecture of transformers to learn bidirectional representations of text, allowing them to better understand the context of words within a sentence (Pérez et al., 2022). BERT can capture the emotional essence of text data and accurately predict the sentiment behind the words. In this research, we have used the BERT model RoBERTuito, a pre-trained language model for user-generated text in Spanish, trained on over 500 million tweets (Pérez et al., 2022), since it has exhibited superior performance compared to other models in Spanish (Pérez et al., 2022). Before training a model, a pre-processing of the data should be carried out and BERT models have their own data pre-processing techniques. Next, the data must be prepared to be accepted by the model, using the Tokenizer class, which converts text inputs into numerical data. To train the models, Transformers provide a Trainer class to fine-tune models on a specific data set. Thus, the training parameters on which we will perform the search are learning rate, batch size, and epochs.

6. Results and discussion

This section describes the results of the different classification models. During testing, we evaluate the model using performance metrics like accuracy, precision, recall, and F1-score. The accuracy metric represents the total percentage of correctly classified values, both positive and negative. The precision metric measures what percentage of values that have been classified as positive are actually positive. Recall, also known as the ratio of true positives, is used to find out how many positive values are correctly classified and finally, F1-score combines precision and recall in a weighted manner. To find the best values for the hyperparameters in the classification models to optimize these performance metrics, we must use techniques to tune these hyperparameters. In particular, for the SVM, RF, and Deep learning models we have used the grid search technique, where this type of search tests all possible combinations of values provided in a parameter grid. In the case of BERT models we use the grid search using the Weights & Biases (W&B) platform, which facilitates the tracking and visualization of metrics, parameters, and model training results.

To evaluate the performance of the classification models, a training set and a validation set will be used, using the conventional 70%–30% split, respectively (NLTK Project, 2023). Additionally, we apply the k-fold cross-validation technique, where the model is trained and assessed k times to identify optimal data set partitions. Specifically, k-fold (k=10) cross-validation will be used, i.e., the model is trained and evaluated 10 times. Furthermore, we implement the EarlyStopping technique (NLTK Project, 2023) to prevent over-fitting and ensure the generalization of our classification models. This method specifies an arbitrary number of training epochs and stops training once the model performance stops improving on a validation dataset.

Regarding the software employed in our work, we have developed our classification models using Python (version 3.1) and the Keras (Oneiros, 2023) and TensorFlow (Google Brain Team, 2023) libraries (using in particular `keras 2.6.0` and `keras-preprocessing 1.1.2`). With respect to the hardware computational resources, it is worth noting that all the experiments and evaluation have been run on a personal computer; concretely, on a MAC Book Pro equipped with an Intel Core i7 processor (I7-1068NG7) at 2.3 GHz (4 cores) and 32 GB of memory (3733 MHz LPDDR4X). The computer described runs on macOS Sonoma (version 14.1.1).

6.1. *Data pre-processing and encoding*

Once the corpus and classification models have been selected, the comments need to be cleaned and processed so that the models can understand the texts under the same conditions. This helps to ensure the homogeneity of the words, as social network messages are usually written in a more informal tone and are therefore more prone to variability and misspellings. In the pre-processing phase, the following methods were used:

- **Standardization of letter case:** All letters are converted to a consistent format, lower case in our case, to ensure uniformity, avoid discrepancies, and reduce text variability. This process reduces the number of words and simplifies the vocabulary to be provided to the classification models to improve their performance.
- **Elimination of accents and diacritical marks**: Accents and diacritical characters are stripped from words to reduce variations in word representations, for example, "á" is converted to "a" or "é" to "e". This process also helps to reduce and simplify the vocabulary.
- **Reduction of character repetition**: The aim is to minimize excessive repetition of characters within words, such as turning "Nooooooooooooooo" into "No", thereby simplifying the final vocabulary.
- **Standardization of expressions of laughter**: The objective is to convert laughter expressions or emoticons into a standardized format, allowing models to interpret them consistently.

Table 4
Optimization of the kernel type hyperparameter (SVM model).

Kernel type	Poly	*Rbf*	Sigmoid
Accuracy	48%	*53%*	34%

- **Standardization of slang and jargon**: Specific adjustments are made to standardize some slang or colloquial expressions commonly used in the context of social networks, ensuring uniformity and clarity. As an example, the abbreviation "xo" will be transformed into the word "pero" ("but" in English), or "xfa" will be transformed to "por favor" ("please" in English) thus simplifying the vocabulary.
- **Removal of mentions, hashtags, and URLs (links)**: User mentions, hashtags, and hyperlinks are removed, as they do not provide relevant information. Consequently, the size of the vocabulary is reduced and we focus on the most important words. We also filtered out messages from bots used on Twitch (Streamlabs and Nightbot). Furthermore, to keep the highest level of anonymity, we have anonymized the usernames of all participants in the chats.
- **Deletion of punctuation marks**: Punctuation marks, such as periods, commas, or question marks, are omitted because they do not provide important information to models. The deletion of punctuation makes it possible to eliminate unnecessary symbols and to make the vocabulary cleaner and simpler to be provided to the algorithms.

This is followed by the feature reduction process that will reduce the size of the vocabulary to be used by the classification models. The following techniques have been considered:

- **Delete Stopwords:** Some words, although essential for constructing meaningful sentences, do not carry significant polarity information in texts or sentences. In Spanish, these words typically include prepositions, pronouns, conjunctions, and verbs like "haber" ("to have" in English). This deletion reduces noise and vocabulary size and keeps the focus on the important words.
- **Applying stemming:** Stemming is a process of morphological normalization in which a word is reduced to its root. For example, the result of stemming the word "ganador" ("winner" in English) in Spanish would be "ganad". Stemming reduces the number of words in the text and simplifies the vocabulary to be provided to the classification models to optimize their performance. For this purpose, SnowballStemmer (Python Software Foundation, 2021), widely used in Spanish, will be used.

The BERT models then implement their data pre-processing. In the case of RoBERTuito: Character repetitions are limited to three, usernames are converted to a special token, hashtags are replaced by a special token, and emojis are replaced by their textual representation. However, RoBERTuito was tested with both data pre-processing techniques (RoBERTuito data pre-processing and the previous described) and the performance in both sets of results were very similar.

After the pre-processing step, messages are tokenized, which converts our character sequences, words, or paragraphs into computer inputs. This process divides a text into smaller units called tokens, words in our case. Thus, the token can be thought of as the unit for semantic processing. In our work, we use the TweetTokenizer (Witten et al., 2011) tokenizer for all models except for the BERT (RoBERTuito) models, which implement their own tokenizers.

The next step is to extract features from tokens, which transforms text into numbers, since classification models cannot interpret text. Specifically, both the text message, already normalized and tokenized, and the label, corresponding to the polarity/emotion associated with each message, must be transformed. To transform messages into numbers, a dictionary has been created in which each message is represented by a corresponding index vector in the dictionary (word embedding). One-hot encoding (commonly used in text classification) is used, where categorical features are converted into a numerical form to be processed by machine learning algorithms. As we are categorizing into three polarities (P, N, and NEU), we use three columns. In this setup, a "1" is assigned to the column corresponding to the associated polarity class of the message, while "0" is assigned to the other two columns. However, in the SVM and RF models, a subtle modification has been made to the one-hot encoding. Instead of utilizing a separate column for each variable and assigning binary values (0 or 1), a single encompassing variable (output) has been introduced. Regarding polarity, it is now represented as either P, N, or NEU, in contrast to the previous binary coding of 0 or 1.

6.2. Results of the polarity response

6.2.1. Results of the SVM model

The best values of the hyperparameters of the SVM classification model are optimized starting with the kernel type and then the regularization parameter. Table 4 shows that the RBF kernel obtains the best accuracy results, reaching 53%. Next, we search for the best value for the regularization parameter (Table 5), finding the optimal value at C=150, but observing that the accuracy values hardly increase, achieving a maximum value of 55%.

Other than analyzing accuracy, it is necessary to evaluate the model in each class separately using the Precision, Recall, and F1-score metrics (Table 6). These results show that the N class is the best-predicted class, reaching an accuracy of 60%, although the Recall and F1-score perform worse. In addition, the NEU class performs very poorly on all metrics.

Table 5
Optimization of the regularization hyperparameter (SVM model).

C	0.01	0.001	0.1	1	10	50
Accuracy	46%	46%	50%	53%	54%	54%
C	*150*	200	300	500	400	10 000
Accuracy	*55%*	54%	54%	53%	53%	50%

Table 6
Summary of the SVM model results for Precision, Recall, and F1-score metrics considering three polarity classes (P, N, NEU).

	Precision	Recall	F1-score
P	54%	80%	64%
N	60%	40%	48%
NEU	32%	11%	16%

Table 7
Optimization of the kernel type hyperparameter (RF model).

Max_features	Log2	*Sqrt*
Accuracy	66%	*69%*

Table 8
Optimization of the number of trees hyperparameter (RF model).

Number of trees	10	100	200	300
Accuracy	66%	67%	68%	69%
Number of trees	*500*	700	1000	10 000
Accuracy	*70*%	69%	69%	66%

Table 9
Summary of the RF model results for Precision, Recall, and F1-score metrics considering three polarity classes (P, N, NEU).

	Precision	Recall	F1-score
P	71%	78%	74%
N	68%	73%	70%
NEU	78%	33%	47%

6.2.2. Results of the RF model

The hyperparameters of this model are optimized starting with the maximum number of predictors (max_features) and continuing with the number of trees considered. Thus, the best value for the first hyperparameter is *sqrt*, which increases the accuracy up to 69%, as shown in Table 7. Next, we proceed to find the best value for the number of decision trees (Table 8). As shown, the number of decision trees does not have a significant impact on the accuracy. However, the optimal value is reached for 500 trees, with an accuracy of 70%.

Table 9 shows the results of Precision, Recall, and F1-score for each class. The P class is the best-predicted class, as their metrics show the best results, all being above 70%. The N class shows a precision of 68% with better results in the recall and F1-score metrics. The NEU class also shows good results on precision but fails badly on recall and F1-score. This result is quite rational, as it is quite difficult to detect neutral comments when expressing opinions on social networks. Finally, it can be concluded that RF significantly outperforms SVM in all metrics.

6.2.3. Results of the deep learning model

To train and assess the model's performance, various tests have been conducted to fine-tune all hyperparameters. These adjustments are outlined in the following order of analysis: Tuning the number of filters and neurons, optimizing dropout rates and the learning rate for the Adam optimizer, reducing the total number of words in the corpus, and lastly, fine-tuning the batch size.

Hidden layers (Goodfellow et al., 2016) are intended to improve the model's ability to extract important features, thus achieving a more accurate classification. For the tests, three layers of size 256, 128, and 64 neurons were used simultaneously. The position chosen to place these hidden layers just after the LSTM layer is to form a hierarchical representation in which the convolution layer is responsible for capturing local and low-level features, while the hidden layers are responsible for capturing global and high-level features. Placing them in this way allows the model to learn more abstract representations, which is a great advantage in NLP problems. In this particular work, a simultaneous study of the results was carried out both with and without hidden layers, obtaining an improvement of 10% by integrating the hidden layers. Table 10 shows the metric values obtained for the configuration with the highest accuracy rate being 65%.

Table 10
Results obtained to assess number of neurons and filters (Deep Learning model).

Number filters Convolutional layer	Number neurons LSMT layer	Neurons in hidden layers	Dropout parameter	Recurrent parameter	Kernel size	Cross-Validation Accuracy
192	256	256, 128, 64	0.2	0.3	8	55.06%
192	*128*	*256, 128, 64*	*0.2*	*0.3*	*8*	*65%*
192	96	256, 128, 64	0.2	0.3	8	63.85%
192	64	256, 128, 64	0.2	0.3	8	55.06%
180	256	256, 128, 64	0.2	0.3	8	64.51%
180	96	256, 128, 64	0.2	0.3	8	52.40%
160	128	256, 128, 64	0.2	0.3	8	52.74%
160	64	256, 128, 64	0.2	0.3	8	59.87%
150	256	256, 128, 64	0.2	0.3	8	51.41%
128	128	256, 128, 64	0.2	0.3	8	62.69%
128	64	256, 128, 64	0.2	0.3	8	54.89%
96	64	256, 128, 64	0.2	0.3	8	53.90%
192	150	256, 128, 64	0.2	0.3	8	53.90%

Table 11
Results obtained for dropout rate variation (Deep Learning model).

Number filters Convolutional layer	Number neurons LSMT layer	Neurons in hidden layers	Dropout parameter	Recurrent parameter	Kernel size	Cross-Validation Accuracy
192	*128*	*256, 128, 64*	*0.2*	*0.3*	*8*	*65%*
192	128	256, 128, 64	0.3	0.3	8	53.07%
192	128	256, 128, 64	0.4	0.3	8	55.22%
192	128	256, 128, 64	0.5	0.3	8	55.39%
192	128	256, 128, 64	0.6	0.3	8	56.88%
192	128	256, 128, 64	0.7	0.3	8	54.73%
192	128	256, 128, 64	0.8	0.3	8	58.21%
192	128	256, 128, 64	0.2	0.4	8	52.74%
192	128	256, 128, 64	0.2	0.5	8	64.84%
192	128	256, 128, 64	0.2	0.6	8	54.39%
192	128	256, 128, 64	0.2	0.7	8	53.23%
192	128	256, 128, 64	0.2	0.8	8	53.90%

Table 12
Accuracy achieved for different values of learning rate (Deep Learning model).

Learning rate	0.001	0.003	*0.005*	0.007	0.009	0.01
Cross-Validation Accuracy	65.01%	65.01%	*65.5%*	61.69%	64.34%	64.84%

Table 13
Accuracy achieved for different values of vocabulary sizes (Deep Learning model).

Vocabulary size	All	1653	1553	1453	*1253*	1153
Cross-Validation Accuracy	65.51%	68.16%	70.32%	70.81%	*72.14%*	69.98%

Table 14
Accuracy achieved for different values of Batch size (Deep Learning model).

Batch Size	32	64	128	*256*	512
Cross-Validation Accuracy	71.80%	72.14%	73.13%	*73.6%*	72.14%

The next step is to set the dropout and recurrent dropout rates of the LSTM layer. According to the Keras documentation (Oneiros, 2023), these dropout rates can be modified between 0 and 1. Table 11 shows that this parameter causes significant changes in the model accuracy, and the best performance, 65%, is achieved with the values 0.2 and 0.3.

The next step is to optimize the learning rate of the Adam optimizer. This parameter is essential both to control the speed of convergence and to overcome local minima that may occur. As shown in Table 12, there is no considerable variation in the model's accuracy metric when the value of this parameter is changed. The highest accuracy is 65.5% with a learning rate of 0.005.

The next phase of training is to optimize the total number of words in the corpus used by the model to improve the efficiency. There is a strong trade-off between decreasing the vocabulary size and the possible loss of critical information. In this analysis, we have examined the performance by varying the number of words in increments of 100. As Table 13 shows, reducing the number of words by excluding the least relevant terms improves the accuracy of the model. However, above a certain threshold (around

Table 15
Summary of the final values for the optimized model.

Parameters	Values
Number of filters at the convolutional layer	192
Number of neurons at the LSTM layer	128
Number of neurons of hidden layer	256, 128, 64
Dropout rate	0.2
Recurrent dropout rate	0.3
Learning rate	0.005
Vocabulary size	1253
Batch Size	256

Table 16
Summary of the hybrid deep learning model results for Precision, Recall, and F1-score metrics considering three polarity classes (P, N, NEU).

	Precision	Recall	F1-score
P	73%	71%	72%
N	75%	84%	79%
NEU	63%	35%	45%

Table 17
Summary of RoBERTuito results for Precision, Recall, and F1-score metrics considering three polarity classes (P, N, NEU).

	Precision	Recall	F1-score
P	82.3%	85.6%	84%
N	75%	85%	80%
NEU	52.5%	18.4%	27%

Table 18
Comparison of all algorithms for the Precision metric.

RoBERTuito	Deep learning	RF	SVM
78%	73.3%	70%	60%

1150 words), further reduction decreases accuracy, as certain essential words in the corpus are omitted. In our case, the number of words achieving the best accuracy, 72.14%, is 1253.

Finally, the batch size parameter defines the amount of data available to the model in each iteration. Values that are too high can lead to excessive demand on memory and resources, but too small values can lead to greater variability in accuracy. The values in Table 14 are typical, and there are no major differences in accuracy, with the best result obtained for a value of 256 with a global accuracy of 73.6%.

The final parameter configuration used in the experiments is shown in Table 15, while Table 16 presents the metric results for each separate class (P, N, NEU), that is, precision, recall and F1-score. The findings reveal that the N class exhibits the highest performance, followed by the P class, while the NEU class shows the lowest performance. Notably, the model encounters the greatest challenge in accurately classifying messages with a neutral polarity. This challenge can be attributed to the difficulty of complete neutrality of messages, compounded by the imbalance of classes within the dataset, where the proportion of cases of neutral polarity is significantly lower than other polarities.

6.2.4. Results of the BERT model: RoBERTuito

We tested the RoBERTuito model with the following parameter settings to obtain the best performance on the global accuracy metric: Learning rate = 9.553×10^{-6}, train batch size = 16, eval batch size = 16, num train epochs = 10.

For this configuration, an overall accuracy of 78% is achieved. If we analyze the performance of the model for each class (precision, recall, F1-score), we can see in Table 17 a rather poor result for the NEU class compared to the P and N polarities. These results may be due to the lack of balance in the dataset, where only 11% of the instances correspond to this class. In contrast, the results for all quality metrics for P and N polarities are quite good, with precision levels of 82% and 75%, improving performance over previous algorithms. Additionally, better results are observed in all metrics for the P versus the N class, with differences close to 7%.

6.2.5. Comparison of the results of the algorithms

Table 18 shows the comparison of accuracy between algorithms. It can be seen that the best performance is achieved with the RoBERTuito model, followed by the Deep Learning model. The lowest performance is obtained by the SVM, as expected, and the RF model provides relatively good results at around 70%.

Table 19
Comparison of all algorithms for Precision, Recall, and F1-score metrics, considering three polarity classes.

Classification model	N			P			NEU		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
RoBERTuito	75%	85%	80%	82.3%	85.6%	84%	52.5%	18.4%	27%
Deep Learning	75%	84%	79%	73%	71%	72%	63%	35%	45%
RF	68%	73%	70%	74%	92%	82%	78%	33%	47%
SVM	60%	40%	48%	54%	80%	64%	32%	11%	16%

Table 20
Training time in seconds of all classification models for polarity.

RoBERTuito	Deep learning	RF	SVM
18,780 (5 h and 13 min.)	90	5	5

It is essential to note that, as mentioned above, classification models in the area of sentiment analysis on social media platforms such as Twitch face significant challenges. These challenges and difficulties, described in Section 1, are even more accentuated in specific environments, such as video games, where language takes on particularly complex characteristics. Specifically, the use of game-specific jargon, extensive use of abbreviations, acronyms, inside jokes, short-lived memes and trends, excessive use of slang (informal English usage), or the use of emotes and emoticons. Despite this challenging context, it is observed that the RoBERTuito model achieves a high accuracy of 78%, while the deep learning and RF models reach values around 70%.

Table 19 shows the values of the precision, recall, and F1-score metrics to compare the performance of each metric across all algorithms. The best performance is shown by RoBERTuito, followed by the hybrid deep learning algorithm, while the lowest performance is achieved by the SVM model. In addition, when comparing all algorithms, some common features appear. Firstly, all algorithms show poor performance in predicting the NEU class, which may be due to two main reasons: It is the minority class (around 11%), which may negatively affect the model's prediction performance, and NEU messages may often lack clear emotional indicators, such as specific words associated with positive or negative emotions. Furthermore, it can be observed that the P and N classes perform best. The results for both classes are quite comparable in the deep learning model, although the P class is better predicted in RoBERTuito and RF. It is important to note that these two classes are the majority classes and their distribution in the dataset is comparable. Finally, it is worth emphasizing that, in situations where the repercussions of false positives or false negatives have significant adverse consequences, as in the case of identifying negative comments on Twitch channels (expressing anger, hate, or disapproval), achieving a high precision score holds paramount importance. Consequently, it can be inferred that RoBERTuito, together with deep learning algorithms, and even RF, not only excel in terms of precision for both classes (P, N) but also demonstrate good performance on recall and F1-score metrics. This underlines the efficiency of these algorithms in effectively detecting extreme polarities.

In addition, the confusion matrix (Fig. 3) helps to evaluate the classification algorithms, allowing us to visualize the performance by comparing the model predictions with the actual result on a test dataset. The confusion matrix provides information on the number of hits and misses of the model in the different categories or classes. The rows represent the real values of the classes, while the columns represent the classes predicted by the model. Comparing the confusion matrix of the two best algorithms, RoBERTuito, and deep learning, RoBERTuito shows less confusion between the three classes, demonstrating better performance in all metrics. Moreover, the deep learning model has a significant confusion of messages whose polarity is N but mistakenly confusing them as those of P polarity, and vice versa, with P messages mistaken as N ones. This behavior also occurs for the RoBERTuito model but to a lesser extent. Analyzing the confusion matrix of the worst-performing algorithms, SVM and RF, it is observed that many negative messages are confused with positive comments, especially for the SVM model. The same is true for positive messages, which are confused with negative comments, but to a lesser extent for both algorithms. In addition, all models perform worse in the no sentiment class (NEU), as the percentage of this class in the data set is very low. It is important to balance the data set, as when having so few neutrally classified texts, as is in this case, the models do not properly learn about the neutral classification. In other words, the models do not correctly learn patterns that indicate the relationship to this class.

Finally, we evaluated the complexity of the algorithms by measuring the training time of each algorithm to compare its complexity versus its accuracy. Table 20 shows the training time (in seconds) for each classification model for polarity. As can be observed, the RF and SVM classification models require very little training time, although their performance is relatively lower, especially that of SVM. In contrast, RoBERTuito takes around five hours of training, which is much longer than the second best algorithm (deep learning) that only requires about 90 s. However, the performance gain of about 6% is quite significant in natural language processing, and considering that the training is only executed once, we consider that this computational time is acceptable considering the gain. It is worth highlighting that, as stated at the beginning of this section, all the training has been conducted on a personal computer; if more powerful resources were available, the training time would drastically decrease, and would therefore be beneficial to opt for RoBERTuito in favor of higher accuracy rates.

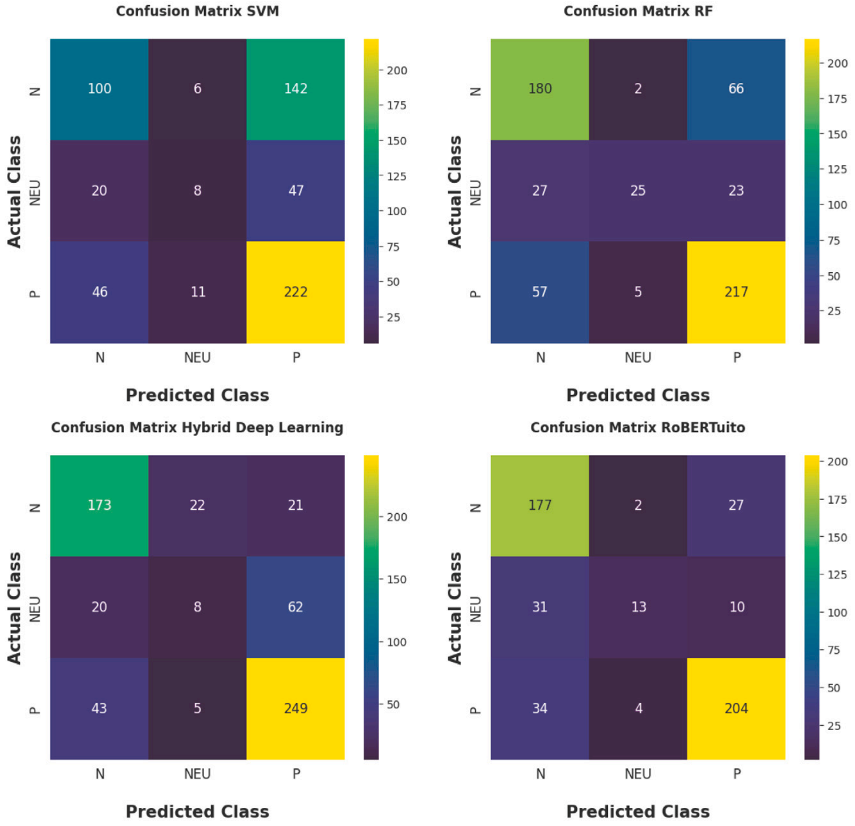


Fig. 3. Confusion matrix of all classification models for the 3 considered classes P, N, NEU.

Table 21
Summary of SVM model results for Precision, Recall, and F1-score metrics considering emotions.

Emotions	Precision	Recall	F1-score
Approval	39%	69%	50%
Hype	43%	21%	28%
Disapproval	31%	27%	29%
Sadness	24%	16%	19%
Anger	20%	8%	11%
Neutral	45%	21%	29%

6.3. Results of the emotional response

6.3.1. Results of the SVM model

The optimal hyperparameters of this model were searched in the same order as for the previous polarity prediction case, i.e. search for the kernel type and then the regularization parameter. The best values were obtained for *RBF* and 1000, respectively, reaching a global accuracy of 37%. If we analyze the performance of each class (Table 21), the best performance corresponds to the positive emotions, Approval and Hype. On the contrary, the lowest performance corresponds to negative emotions, where the emotion of Anger shows the worst result.

6.3.2. Results of the RF model

Again, the search for optimal values for the hyperparameters of this model starts with the number of predictors (max_features) followed by the number of trees. The maximum global accuracy achieved by this model is 49% for a value of sqrt (max_features) and 500 trees. If we analyze the performance of the classes separately (Table 22), RF improves the precision in all emotions compared to SVM, including the Neutral class. One more time, Anger obtains the lowest performance since the recall and F1-score show low performance.

Table 22
Summary of RF model results for Precision, Recall, and F1-score metrics considering emotions.

Emotions	Precision	Recall	F1-score
Approval	49%	63%	55%
Hype	58%	41%	48%
Disapproval	35%	51%	41%
Sadness	57%	29%	39%
Anger	50%	18%	26%
Neutral	75%	51%	61%

Table 23
Summary of the final values of the parameters optimizing the Deep Learning model.

Parameters	Values
Number of neurons at the convolutional layer	180
Number of neurons at the LSTM layer	256
Dropout rate	0.2
Recurrent dropout rate	0.6
Learning rate	0.001
Vocabulary size	2000
Batch Size	32

Table 24
Summary of Deep Learning model results for precision, recall and F1-score metrics considering emotions.

Emotions	Precision	Recall	F1-score
Approval	59%	77%	66%
Hype	62%	54%	58%
Disapproval	43%	59%	49%
Sadness	91%	48%	63%
Anger	57%	16%	25%
Neutral	40%	25%	30%

Table 25
Summary of RoBERTuito model results for precision, recall and F1-score metrics considering emotions.

Emotions	Precision	Recall	F1-score
Approval	76%	76.5%	75.7%
Hype	74.7%	67.5%	70%
Disapproval	59.7%	65.3%	62%
Sadness	85%	93%	89%
Anger	59%	56.5%	56%
Neutral	52.3%	39.7%	44%

6.3.3. Results of the deep learning model

Table 23 shows the value of hyperparameters that allowed optimizing the model for emotion prediction. The process followed was the same as for polarity prediction. This configuration of the hybrid deep learning model achieves an overall accuracy of 55%. A detailed analysis as in the case of polarity has been removed for brevity.

Analyzing the results for emotions classes (Table 24), it can be observed that predictions improve for previous models (SVM, RF). Once again, the Anger emotion and Neutral categories show the lowest performance, especially for the recall and F1-score metrics. In contrast, positive emotions (Approval, Hype) improved their performance along with the emotion Sadness.

6.3.4. Results of the BERT model: RoBERTuito

We tested the RoBERTuito model with the following parameter settings, which allow for the best performance of the global accuracy metrics: Learning rate = 5.15×10^{-5}, train batch size = 4, eval batch size = 16, num train epochs = 9. This configuration achieves a global accuracy of 68%.

If we analyze the performance for each emotion (Table 25), the lowest result is again obtained for the Neutral emotion. Similarly, the positive classes achieve good results and increase their efficiency with respect to previous algorithms, showing precision results between 74%–76% for both classes. The Sadness class achieves the best performance in all the metrics analyzed, with a precision of 85%, achieving even better results for the recall (93%) and F1-score (89%) metrics. In contrast, Anger again obtains the lowest performance among the five emotions, although it improves its effectiveness on all metrics compared to the previous algorithms.

Table 26
Comparison of all algorithms for the accuracy metric.

RoBERTuito	Deep learning	RF	SVM
68%	55%	49%	37%

Table 27
Training time in seconds of all classification models for emotions.

RoBERTuito	Deep learning	RF	SVM
29,400 (8 h and 10 min)	180	5	5

6.3.5. *Comparison of the results of the algorithms*

Table 26 shows the comparison of the global accuracy of the four algorithms. Once again, the best performance corresponds to the RoBERTuito model (around 68%) followed by the Deep Learning model (around 55%). In contrast, SVM has the lowest performance with an accuracy of only 37%, as expected. In this case, the results are worse than for the polarity prediction, since, on one hand, we have a wider range of categories (six classes). On the other hand, predicting emotions in social networks is a challenge due to certain factors such as: Variability in emotional expression, as people express their emotions in different ways, and may vary between individuals; subjectivity as the interpretation of emotions is inherently subjective (what one person considers as an expression of sadness, another may see as something different); or even emotional expressions may vary according to the culture and community of the user. Nevertheless, good values close to 70% are achieved with the RoBERTuito algorithm.

Table 28 shows the values of the precision, recall, and F1-score metrics when predicting each emotion separately. The performance of the algorithms varies depending on the emotion being detected. Regarding the Approval emotion, generally, all the metrics for this emotion are relatively high compared to other emotions, with RoBERTuito leading in precision and F1-score. It is also worth noting that it is the majority class within the dataset. In contrast, Anger shows the lowest F1-score overall in all algorithms, indicating significant difficulties in detecting this emotion. However, it should be noted that this class is the minority class in the dataset, which may negatively affect its performance. For the Disapproval class, RoBERTuito shows the best balance between precision and recall concerning the other algorithms, although deep learning also has a reasonably high recall, but its precision is lower than the RoBERTuito model precision. The same behavior is observed for the Hype class, where RoBERTuito shows the strongest performance with the highest precision, recall, and F1-score, followed by the deep learning algorithm with a good performance. Finally, for the Sadness class, the RoBERTuito algorithm again leads the ranking, although deep learning still shows relatively good precision and F1-score values as well.

The confusion matrix (Fig. 4) also shows the performance of every algorithm by comparing the model predictions with the actual results. Comparing the two best algorithms, RoBERTuito, and deep learning, RoBERTuito shows less confusion between all emotional classes than the deep learning algorithm. The greatest confusion in both algorithms appears between the emotions of Approval and Disapproval, being more important in the deep learning algorithm. Analyzing SVM and RF, both algorithms show more confusion between all emotional categories than RoBERTUito and deep learning, particularly in the Approval emotion, which is wrongly predicted with other emotions. Moreover, RF performs better than SVM and RF shows the same behavior as RoBERTuito and deep learning between the Approval and Disapproval emotions, although the confusion of this class in RF is higher.

Finally, we have measured the training time of each algorithm to analyze its complexity. Table 27 shows that once again RF and SVM show a shorter training time, although their performance in emotion detection is very low, especially SVM with an accuracy capability of only 37%. In contrast, RoBERTuito has the longest training time, well above the deep learning algorithm, which requires about 180 s. However, in the case of emotion detection, the performance gain of RoBERTuito is much more significant, around 13%, and since the training is done only once, this computational time is quite affordable compared to the performance improvement.

In summary, Anger is the worst predicted emotion, while Approval, Hype, and Sadness tend to be the best predicted, a fact that coincides with the most minority class and the two most majority classes, respectively. RF and SVM are the poorest performing algorithms, although RF shows intermediate performance in some categories (Sadness and Hype). Finally, RoBERTuito and Deep Learning often perform well in all categories, especially RoBERTUito, but their performance varies depending on the specific emotion. Indeed, the deep learning model tends to have good recall in general, while RoBERTuito shows solid precision on several emotions.

7. Conclusion

The research presented in this paper is the first to focus on analyzing the emotional response to video games on Twitch (the most popular platform used among gamers worldwide) applying machine learning techniques. The main research objectives were twofold: Design and provide a novel corpus, and explore the usage of Machine Learning algorithms to detect emotional response, polarity, and emotions in Spanish Twitch channels broadcasting video game content.

First, a corpus has been designed based on messages from Spanish video game streamers' channels on Twitch and manually labeled with polarity and emotions. This corpus is available online for researchers to use in video game contexts or artificial intelligence applications. Importantly, this is the first corpus in Spanish built to analyze the impact of social response on Twitch gaming channels and the first time that sentiment analysis has been applied to detect emotional responses in such environments.

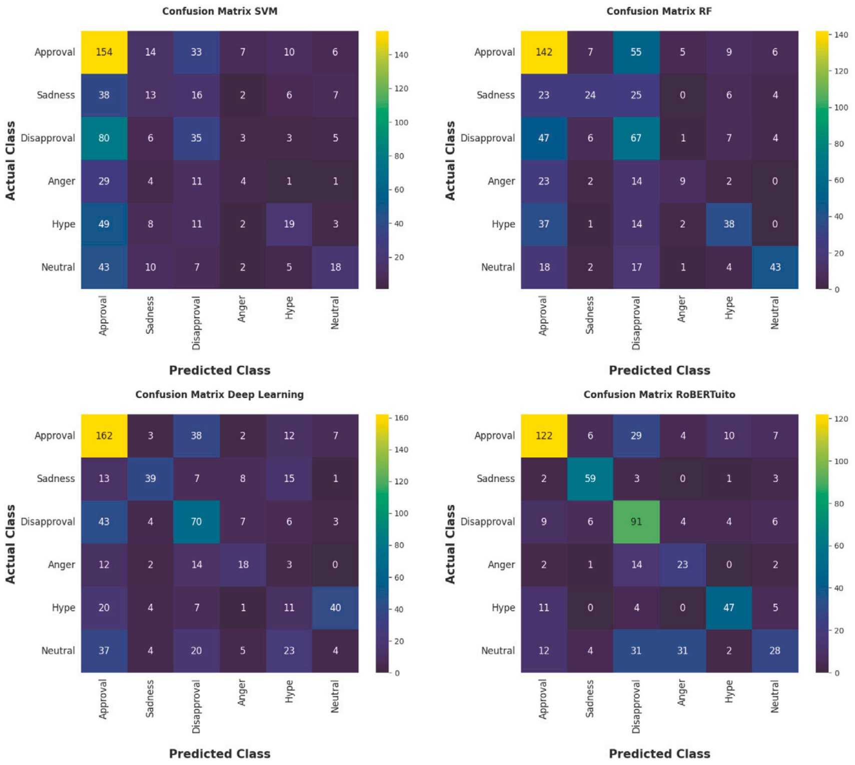


Fig. 4. Confusion matrix of all classification models (SVM, RF, RoBERTuito, deep learning) for emotions.

Table 28
Comparison of RoBERTuito, Deep Learning, RF and SVM algorithms for precision, recall and F1-score considering emotions.

Emotions	Metric	RB	DL	RF	SVM
Approval	Precision	76%	59%	49%	39%
	Recall	76.5%	77%	63%	69%
	F1-score	75.7%	66%	55%	50%
Hype	Precision	74.7%	62%	58%	43%
	Recall	67.5%	54%	41%	21%
	F1-score	70%	58%	48%	28%
Disapproval	Precision	59.7%	43%	35%	31%
	Recall	65.3%	59%	51%	27%
	F1-score	59.7%	49%	41%	29%
Sadness	Precision	85%	91%	57%	24%
	Recall	93%	48%	29%	16%
	F1-score	89%	63%	39%	19%
Anger	Precision	59%	57%	50%	20%
	Recall	56.5%	16%	18%	8%
	F1-score	56%	25%	26%	11%
Neutral	Precision	52.3%	40%	75%	45%
	Recall	39.7%	25%	51%	21%
	F1-score	44%	30%	61%	29%

Furthermore, our research also addresses the application of machine learning algorithms to enable the detection of the emotional response generated on streamers' channels to video games. In this way, it is necessary to emphasize that sentiment analysis in virtual environments, in this case Twitch, faces very significant challenges. Despite this adverse context, the RoBERTuito classification model achieves a high accuracy of 78% in detecting polarity, while deep learning and RF models achieve values around 70%. Regarding emotion detection, the best performance belongs to RoBERTuito (around 68%) followed by the deep learning model (55%). These good results prove that our first hypothesis is true, since our machine learning algorithms are able to efficiently classify Twitch chat messages with polarity and emotions. Although accuracy levels are not as good as in polarity detection, it is important to note that emotion detection is more difficult due to multiple factors, including the fact that people express their emotions in different ways and that there is a strong subjectivity in the interpretation of emotions. All this is combined with the complexity of language in the context of video games on virtual platforms. However, this also serves to validate our second hypothesis, as our proposal helps to measure the emotional impact of Twitch streaming sessions through the analysis of chat messages, determining the most predominant emotions.

In summary, the incorporation of machine learning techniques in the gaming environment and Twitch is innovative, as it allows to identify emotional responses in the streamers' channels. The audience engagement related to their emotions has been analyzed in previous works by considering facial expression detection (Teixeira et al., 2012; Mishra et al., 2022) or positive and negative emotions induced by advertisements on YouTube (Kujur and Singh, 2018), for example. Therefore, we believe that our research work could be used in the context of video games and Twitch in a similar way by psychology and marketing experts to offer benefits such as helping streamers understand viewers' feelings about games, streams, or content, leading to better audience retention. It could also help to offer personalized content recommendations based on emotions and help streamers improve their content. Lastly, this research can be used to detect toxic or inappropriate behavior in Twitch chats following strategies such as those employed in other different contexts, as described in the studies (Risch and Krestel, 2020) (which focuses on general online discussions) and Taleb et al. (2022) (that covers social media platforms).

CRediT authorship contribution statement

Noemí Merayo: Formal analysis, Data curation, Conceptualization, Investigation, Methodology, Project administration, Software, Supervision, Validation, Writing – original draft, Writing – review & editing. **Rosalía Cotelo:** Conceptualization, Data curation, Investigation, Methodology, Supervision, Visualization, Writing – original draft, Writing – review & editing. **Rocío Carratalá-Sáez:** Conceptualization, Formal analysis, Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Francisco J. Andújar:** Data curation, Formal analysis, Investigation, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data available at: https://github.com/noemeralv/Corpus-Twitch-Videogames.

Acknowledgments

This work has been supported by the University of Valladolid, Spain.

References

Agca, Yilmaz, 2023. Analyzing video game content and sentiment a study on categories, emotional responses, and success factors. In: International Research in Social, Human and Administrative Sciences XIV. pp. 67–79.

Plaza del Arco, Flor Miriam, Strapparava, Carlo, Urena Lopez, L. Alfonso, Martin, Maite, 2020. EmoEvent: A multilingual emotion corpus based on different events. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, pp. 1492–1498.

Barbieri, Francesco, Espinosa-Anke, Luis, Ballesteros, Miguel, Soler-Company, Juan, Saggion, Horacio, 2017. Towards the understanding of gaming audiences by modeling twitch emotes. In: Proceedings of the 3rd Workshop on Noisy User-Generated Text. Association for Computational Linguistics, Copenhagen, Denmark, pp. 11–20.

Cai, Jie, Chowdhury, Sagnik, Zhou, Hongyang, Wohn, Donghee Yvette, 2023. Hate raids on twitch: Understanding real-time human-bot coordinated attacks in live streaming communities. Proc. ACM Hum.-Comput. Interact. 7 (CSCW2).

Cai, Jie, Guanlao, Cameron, Wohn, Donghee Yvette, 2021. Understanding rules in live streaming micro communities on twitch. In: Proceedings of the 2021 ACM International Conference on Interactive Media Experiences. IMX '21, Association for Computing Machinery, New York, NY, USA, pp. 290–295.

Casado, Javier, Peña-Acuña, Beatriz, 2022. Léxico de videojuegos incluido en la lengua española: un estudio de caso múltiple [Video games lexicon included in Spanish language: A multiple case study]. Linguo Didáctica 1, 15–35.

Chollet, Francois, 2021. Deep learning with Python. Simon and Schuster.

Clement, J., 2023. Online gaming - statistics & facts. https://www.statista.com/topics/1551/online-gaming/#topicOverview. (Accessed 16 October 2023).

Cotelo García, Rosalía, 2022. Jelou pipol: Computer-mediated communication among Spanish-speaking gamers on Twitch. Internet Pragmat. 5 (2), 257–290.

Deng, Jie, Cuadrado, Felix, Tyson, Gareth, Uhlig, Steve, 2015. Behind the game: Exploring the twitch streaming platform. In: 2015 International Workshop on Network and Systems Support for Games. NetGames, pp. 1–6.

Ekman, Paul, 1999. Basic emotions. In: Handbook of Cognition and Emotion. John Wiley & Sons, Ltd, pp. 45–60.

Ford, Colin, Gardner, Dan, Horgan, Leah Elaine, Liu, Calvin, tsaasan, a.m., Nardi, Bonnie, Rickman, Jordan, 2017. Chat speed OP PogChamp: Practices of coherence in massive twitch chat. In: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems. CHI EA '17, Association for Computing Machinery, New York, NY, USA, pp. 858–871.

Gómez García, Salvador, et al., 2007. Videojuegos: El desafío de un nuevo medio a la comunicación social [Video games: The challenge of a new medium to social communication]. Historia Comunicación Soc. 12, 71–82.

Goodfellow, Ian, Bengio, Yoshua, Courville, Aaron, 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Google Brain Team, 2023. Tensor flow library. https://www.tensorflow.org. (Accessed 26 September 2023).

Gros, Daniel, Wanner, Brigitta, Hackenholt, Anna, Zawadzki, Piotr, Knautz, Kathrin, 2017. World of streaming. Motivation and gratification on twitch. In: Meiselwitz, Gabriele (Ed.), Social Computing and Social Media. Human Behavior. Springer International Publishing, Cham, pp. 44–57.

Guzsvinecz, Tibor, Szűcs, Judit, 2023. Length and sentiment analysis of reviews about top-level video game genres on the Steam platform. Comput. Hum. Behav. 149, 107955.

Hallgren, Kevin A., 2012. Computing inter-rater reliability for observational data: An overview and tutorial. Tutor. Quant. Methods Psychol. 8 (1), 23–24.

Hämäläinen, Mika, Alnajjar, Khalid, Poibeau, Thierry, 2022. Video games as a corpus: Sentiment analysis using Fallout New Vegas dialog. In: Proceedings of the 17th International Conference on the Foundations of Digital Games. FDG '22, Association for Computing Machinery, New York, NY, USA.

Hamilton, William A., Garretson, Oliver, Kerne, Andruid, 2014. Streaming on twitch: Fostering participatory communities of play within live mixed media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '14, Association for Computing Machinery, New York, NY, USA, pp. 1315–1324.

Heise, Anne Hove Henriksen, Hongladarom, Soraj, Jobin, Anna, Kinder-Kurlanda, Katharina, Sun, Sun, Lim, Elisabetta Locatelli, Markham, Annette, Reilly, Paul J., Tiidenberg, Katrin, Wilhelm, Carsten, 2020. Internet research: Ethical guidelines 3.0. https://aoir.org/reports/ethics3.pdf. (Accessed 25 October 2023).

Jenkins, Henry, 2009. Fans, blogueros y videojuegos: la cultura de la colaboración [Fans, bloggers and video games: the culture of collaboration], vol. 180, Grupo Planeta (GBS).

Jodén, Henrik, Strandell, Jacob, 2022. Building viewer engagement through interaction rituals on twitch.tv. Inf. Commun. Soc. 25 (13), 1969–1986.

Kujur, F., Singh, S., 2018. Emotions as predictor for consumer engagement in YouTube advertisement. J. Adv. Manag. Res. 15 (2), 184–197.

Lafrance, Jean Paul, 2003. El juego interactivo: el primer medio de masas de la era electrónica [The interactive game: The first mass media of the electronic age]. Quaderns del CAC 15, 59–68.

Larose, Daniel T., Larose, Chantal D., 2014. Decision trees. In: Discovering Knowledge in Data. John Wiley & Sons, Ltd, pp. 165–186.

Logitech Services S.A., 2014. Streamlabs webpage. https://streamlabs.com/es-es/. (Accessed 17 October 2023).

Merayo, Noemí, Cotelo, Rosalía, Carratalá-Sáez, Rocío, Andújar, Francisco J., 2023. Repository of corpus of video games on Spanish Twitch channels. https://github.com/noemeralv/Corpus-Twitch-Videogames. (Accessed 19 October 2023).

Mishra, Eti, Nikam, Piyush, Vidhyadharan, Sreejith, Cheruvalath, Reena, 2022. An affect-based approach to detect collective sentiments of film audience: Analyzing emotions and attentions. Acta Psychol. 230, 103736.

Moreno-Ortiz, Antonio, Hernández, Chantal Pérez, 2013. Lexicon-based sentiment analysis of Twitter messages in Spanish. Procesamiento del Lenguaje Nat. 50, 93–100.

NightDev, LLC., 2023. Nightbot website. https://nightbot.tv/. (Accessed 17 October 2023).

NLTK Project, 2023. Natural language toolkit (NLTK) library. https://www.nltk.org/. (Accessed 26 September 2023).

Noble, William S., 2006. What is a support vector machine? Nature Biotechnol. 24 (12), 1565–1567.

Olejniczak, Jedrzej, 2015. A linguistic study of language variety used on twitch.tv: Descriptive and corpus-based approaches. In: International Conference RCIC'15 - Redefining Community in Intercultural Context. pp. 329–334.

Oneiros, 2023. Keras library. https://keras.io/. (Accessed 26 September 2023).

Pardo, Lewis, 2023. Twitch downloader repository. https://github.com/lay295/TwitchDownloader. (Accessed 26 September 2023).

Pérez, Juan Manuel, Furman, Damián A., Alemany, Laura Alonso, Luque, Franco, 2022. RoBERTuito: a pre-trained language model for social media text in Spanish.

Plutchik, Robert, 1980. Chapter 1 - A general psychoevolutionary theory of emotion. In: Plutchik, Robert, Kellerman, Henry (Eds.), Theories of Emotion. Academic Press, pp. 3–33.

Pöyhönen, Teemu, Hämäläinen, Mika, Alnajjar, Khalid, 2022. Multilingual persuasion detection: Video games as an invaluable data source for NLP. In: Proceedings of the 2022 DiGRA International Conference. In: Proceedings of Digital Games Research Association, DiGRA, DiGRA 2022 International Conference : Bringing Worlds Together ; Conference date: 07-07-2022 Through 11-07-2022.

Probst, Philipp, Wright, Marvin N., Boulesteix, Anne-Laure, 2019. Hyperparameters and tuning strategies for random forest. Wiley Interdiscip. Rev. Data Min. Knowl. Disc. 9 (3), e1301.

Python Software Foundation, 2021. Snowball algorithms for stemming. https://pypi.org/project/snowballstemmer. (Accessed 26 September 2023).

Ravenbtw, 2023. Twitch database. Global emotes. https://www.twitchdatabase.com/global-emotes. (Accessed 26 September 26 2023).

Rebitzer, Dominik, 2022. Twitch leecher repository. https://github.com/Franiac/TwitchLeecher. (Accessed 26 September 2023).

Recktenwald, Daniel, 2017. Toward a transcription and analysis of live streaming on Twitch. J. Pragmat. 115, 68–81.

Redondo, Jaime, Fraga, Isabel, Padrón, Isabel, Comesaña, Montserrat, 2007. The Spanish adaptation of ANEW (affective norms for English words). Behav. Res. Methods 39 (3), 600–605.

Risch, Julian, Krestel, Ralf, 2020. Toxic comment detection in online discussions. In: Deep Learning-Based Approaches for Sentiment Analysis. Springer Singapore, Singapore, pp. 85–109.

Seering, Joseph, Kairam, Sanjay R., 2022. Who moderates on twitch and what do they do? Quantifying practices in community moderation on twitch. Proc. ACM Hum.-Comput. Interact. 7 (GROUP).

Similarweb LTD, 2023. Website traffic: Check and analyze any website. https://www.similarweb.com/es/. (Accessed 3 October 2023).

Sjöblom, Max, Hamari, Juho, 2017. Why do people watch others play video games? An empirical study on the motivations of Twitch users. Comput. Human Behav. 75, 985–996.

Stephanie, Rennick, Melanie, Clinton, Elena, Ioannidou, Liana, Oh, Charlotte, Clooney, T.E., Edward, Healy, G., Roberts Seán, 2023. Gender bias in video game dialogue. Royal Soc. Open Sci. 10 (5).

Taleb, Mohammed, Hamza, Alami, Zouitni, Mohamed, Burmani, Nabil, Lafkiar, Said, En-Nahnahi, Noureddine, 2022. Detection of toxicity in social media based on natural language processing methods. In: 2022 International Conference on Intelligent Systems and Computer Vision. ISCV, pp. 1–7.

Teixeira, Thales, Wedel, Michel, Pieters, Rik, 2012. Emotion-induced engagement in internet video advertisements. J. Mar. Res. 49 (2), 144–159.

Tretkoff, Ernie, 2008. October 1958: Physicist invents first video game. Adv. Phys. 17.

Twitch Tracker, 2023. Twitch channels, games and Global statistics. https://twitchtracker.com/. (Accessed 3 October 2023).

TwitchTracker, 2024. Twitch language statistics. https://twitchtracker.com/languages.

Twitch.tv, 2023. Terms of service of Twitch.tv. https://www.twitch.tv/p/en/legal/terms-of-service/. (Accessed 25 October 2023).

Velez, John A., Gotlieb, Melissa R., Graybeal, Geoffrey, Abitbol, Alan, Villarreal, Jonathan A., 2018. Live streams and revenue streams: Twitch as a hybrid gaming culture. In: Video Games. Routledge, pp. 193–207.

Vera, José Agustín Carrillo, 2015. La dimensión social de los videojuegos online: de las comunidades de jugadores a los e–sports [The social dimension of online video games: From player communities to e-sports]. Index. comunicación: Revista científica en el ámbito de la Comunicación Aplicada 5 (1), 39–51.

Witten, Ian H., Frank, Eibe, Hall, Mark A., 2011. Data Mining: Practical Machine Learning Tools and Techniques. In: The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Boston.

Yus, Francisco, 2020. La comunicación en la era digital [communication in the digital age]. In: Escandell, M.V., Ahern, A., Amenós, J. (Eds.), Pragmática [Pragmatics]. Akal, Madrid, pp. 608–623.

Zsila, Á., Shabahang, R., Aruguete, M.S., Bőthe, B., Gregor-Tóth, P., Orosz, G., 2023. Exploring the association between Twitch use and well-being. In: Psychology of Popular Media. American Psychological Association.

Noemí Merayo received the Telecommunication Engineer degree from the Valladolid University, Spain, in February 2004 and the Ph.D. degree in the Optical Communication Group at the Universidad de Valladolid, in July 2009. She works as Lecturer at the Universidad de Valladolid. She has also been a Visiting Research Fellow at the University of Hertfordshire in the Optical Networks Group, Science and Technology Research Institute, the TOyBA research group (University of Zaragoza), and the Technology University of Munich (TUM). Her research focuses on the design and performance evaluation of optical networks and the application of artificial intelligence techniques.

Rosalía Cotelo is a Doctor in Spanish Language from the Universidad de A Coruña in 2010. She has been a visiting professor at the University of North Carolina at Chapel Hill, an associate professor at the Universidad Carlos III de Madrid and she is currently a PhD Assistant Professor at the Universidad Autónoma de Madrid. As for her work as a researcher, her interest is divided between Lexicography and the study of the language of young people; the latter from both a lexical and a pragmatic point of view.

Rocío Carratalá-Sáez received a B.Sc. Degree in Computational Mathematics by Universitat Jaume I (UJI) of Castellón (Spain) in 2015, M.Sc. Degree in Parallel and Distributed Computing by Universitat Politècnica de València (Spain) in 2016, and Ph.D. in Computer Science by UJI in 2021. She is currently a postdoc researcher at Universitat Jaume I in the Department of Computer Science and Engineering. Her main research interest is High-Performance Computing, focused on the parallelization of linear algebra operations and scientific applications. More information about her current research activities can be found at https://rociocarratalasaez.es.

Francisco J. Andújar received the M.Sc. degree in Computer Science from the University of Castilla-La Mancha, Spain, in 2010, and the Ph.D. degree from the University of Castilla-La Mancha in 2015. He worked in the Universitat Politècnica de València under a post-doctoral contract Juan-de la Cierva, and currently works in the University of Valladolid as Associate Professor. His research interests include multicomputer systems, cluster computing, HPC interconnection networks, switch architecture, and simulation tools.