IMT 547 Final Project: Phase 1

Team 1: Chesie Yu, Hongfan Lu, Bella Wei

02/20/2024

Project Overview: YouTube Gaming Comments Toxicity

Team name: *Team 1*
Team members: *Chesie Yu, Hongfan Lu, Bella Wei*

Problem Description:
Toxicity in the gaming community is a prevalent problem that hinders the harmonious development of the gaming industry. Our objective is to tackle this concern by exploring whether the game category (Action and Non-Action) serves as a primary determinant of toxicity levels in YouTube video comments. This study focuses on the observational perspective rather than the player angle. If proven, it can offer valuable insights for gaming community management, game design, and the design of social media platforms.

RQ1:

- Do videos of action games arouse significant more toxic comments than non-action games in YouTube?
- Our project is interested in investigating sections for action and non-action based gaming videos on YouTube.
- We hypothesis that there will be different type of emotions represented in the comment section;
- Moreover, the occurance of profanity might be higher in the action based gaming videos.

RQ2:

- Which kinds of gaming video attract most toxic comments? Any pattern behind the scene?
- How does YouTubers' contents, characteristics and behaviors influence toxicity in comments section.

Qualitative:

Question, Problem, Hypothesis, Claim, Context, Motivation

Question:

- Do videos of action games arouse significant more toxic comments than non-action games in YouTube?

- Is gaming category a primary predictor for toxic comments in YouTube?

Hypothesis:

We assume action game(including violence scene) will trigger significant more toxicity in YouTube, and different categories of games present different level of toxicity.

Context:

Online gaming toxicity refers to harmful and negative conversations within the gaming community, presenting a serious issue across social media platforms. This behavior often involves the use of offensive language, insults, harassment, or even threats exchanged among players. It negatively impacts the harmony of the gaming community and social media platforms, fostering resentment and animosity among people and adversely affecting people's mental health, especially teenagers.

Motivation:

Addressing and mitigating online gaming toxicity is crucial for fostering a positive and inclusive gaming culture. If we can identify the main factors and predictors of online gaming toxicity, we can efficiently prevent and regulate this problem, providing valuable insights to gaming industry, gaming design, gaming community management, social media platform, and gamers. It would contribute to creating a healthier and more enjoyable online gaming environment while also aiding in the prevention of cyberbullying and game-related mental health problems.

Definitions, Data, Methods to be Used

Definitions:

Online gaming toxicity refers to the use of harmful and negative language within the gaming community. Our focus was on the YouTube platform, specifically examining comments under gaming videos of each category. Our research novelty lies in that we address this issue from the perspective of gaming observation rather than real-time gameplay.

Data:

We scraped videos under 33 English-Speaking YouTube channels from a list of top 100. For each channel, we select 30 video for action games and 30 for non-action games. We get 140637 comment records in total.

Methods:

In data collection phase we use YouTube API key to get access to chosen YouTube channels. We perform text cleaning on the collected data by removing missing values and unnecessary tokens through tokenization and regular expressions. Subsequently, we use the Perspective API to assess toxicity levels, and leverage NLP sentiment analysis tools such as Vader, TextBlob, and Empath to evaluate the emotional tone of the comments. Finally, we apply data visualization techniques to enhance the presentation of our analysis results.

Assumptions, Biases

Assumptions: YouTube comments are free-speech and not-filtered seriously.

Biases: We use predifined keywords list to fetch videos, which may not be representative enough.
We retrieve 100 comments under each video, provided by the 'most relevant' order on YouTube, which may introduce bias.

Quantitative:

Data Collection Procedure

1. Utilizing **YouTube Data API** to access Youtube comments

2. Keyword selection

Keywords for Action Games:
action_keywords = ["call of duty", "gta", "the last of us", "god of war", "batman", "red dead redemption", "assassin's creed", "star wars jedi", "resident evil", "cyberpunk", "fallout", "tomb raider", "elden ring"]

Keywords for NonAction Games:
nonaction_keywords = ["minecraft", "pokemon go", "just dance", "it takes two", "uncharted", "brawl stars"]

3. Found 33 English-Speaking youtubers from a list of top 100 (ranked by number of subscriber)

4. For each Youtuber, select 30 video for action games and 30 for non-action games

5. Use Relevance (YouTube internal algo) to extract 100 comments each video

Notebook: 01-data-collection

The detailed data collection code and description can be found in `01-data-collection.ipynb`. Here we will only include a brief overview for readability. Our data collection phase is divide into 3 components:

1. **Authentication & Configuration**: Prepared the necessary tools and configuration for our later data collection

- Set up the YouTube API key
- Install and import necessary libraries
- Configure the logging system
- Initialize a client that can communicates with YouTube API

2. **Utility Function**: Encapsulated a series of necessary operations into functions.

- `get_uploads_id(channel_id)` : Fetch the uploads playlist ID for a given YouTube channel.
- `get_video_ids(uploads_id, max_videos=30, keywords=“”)` : Fetch 30 video IDs containing keyword in a given upload playlist.

- `get_video_info(video_ids)` : Fetch necessary video info from a list of YouTube videos.
  Video Features: [“channel_id”, “channel_name”, “video_id”, “video_title”, “video_creation_time”, “video_description”, “video_tags”, “video_viewcount”, “video_likecount”, "video_commentcount"]
- `get_video_comments(video_ids, max_comments=100)` : Fetch comments (100 max each) for a list of videos.
  Comment Features: [“video_id”, “comment_id”, “comment_author_id”, “comment_text”, “comment_time”, “comment_likecount”, "comment_replycount"]
- `get_youtube_data(channel_ids, max_videos=30, max_comments=100, keywords=“”)` : Main function. Fetch videos and comments for a list of channels

3. **Data Collection**

- Set up the parameters(max 100 comments under max 30 videos for each YouTube channel) and keywords list for ‘Action’ and ‘Non-Action’ game category
- Read a self-built document ‘top 100 popular gaming YouTubers’ as our targeted channels, filtering only english-speaking channels
- Collect and combine comment infos for action and non-action games
- Get and save 140637 comments records

Data Processing

Notebook: 02-preprocessing

Our data preprocessing is divided into 3 components:

1. **Data Cleaning**

- **Handle Missings**: **804 rows with 877 missing values; 139833 rows left after removing NA**
- **Convert Data Types**: Convert time-related columns into datetime data type:
  Comments are from a range of time in **('2011-04-22T01:05:52Z', '2024-02-18T20:15:11Z')**
- A total of 1435 unique videos are included

Summary Statistics

```
[5]: # Check the time range
     yt["video_creation_time"].min(), yt["video_creation_time"].max()
```

```
[5]: ('2011-04-22 01:05:52+00:00', '2024-02-19 20:15:00+00:00')
```

```
[6]: # Number of unique channels
     print(f"Number of unique channels: {yt['channel_id'].nunique()}")
```

```
Number of unique channels: 33
```

```
[7]: # Number of unique videos
     print(f"Number of unique videos: {yt['video_id'].nunique()}")
```

```
Number of unique videos: 1420
```

```
[8]: # Print the summary statistics
     yt.describe()
```

[8]:

	video_viewcount	video_likecount	video_commentcount	comment_likecount	comment_replycount
count	1.389960e+05	1.389960e+05	138996.000000	138996.000000	138996.000000
mean	3.742645e+06	1.213770e+05	7021.169537	231.204279	4.082261
std	6.016032e+06	1.691242e+05	11585.725874	1917.918916	26.140869
min	1.158900e+04	1.580000e+02	15.000000	0.000000	0.000000
25%	6.938660e+05	1.930000e+04	860.000000	0.000000	0.000000
50%	1.915267e+06	5.584000e+04	2594.000000	2.000000	0.000000
75%	4.368518e+06	1.439280e+05	8442.000000	17.000000	1.000000
max	1.086792e+08	1.586707e+06	151333.000000	324721.000000	750.000000

2. **Text Preprocessing**

- Filter English comments only
- Do text cleaning by removing unimportant content such as URLs, mentions and stop words.
- Tokenization and save results into the dataframe

Text Cleaning

```
# Function for text preprocessing
def clean(text):
    """
    Performs text preprocessing steps on one document.
    """
    # Convert to lowercase
    text = text.lower()
    # Remove contractions
    text = contractions.fix(text)

    # Remove URLs
    text = re.sub(r"http\S+", "", text)
    # Remove mentions
    text = re.sub(r"(?<![@\w])@(\w{1,25})", "", text)
    # Remove hashtags
    text = re.sub(r"(?<![#\w])#(\w{1,25})", "", text)
    # Remove new line characters
    text = re.sub("\n", " ", text)

    # Remove non-alphabetic characters
    text = re.sub(r"[^a-zA-Z\s]", "", text)

    # Remove extra spaces
    text = re.sub(r"\s+", " ", text)

    # Remove stop words
    stop_words = set(stopwords.words("english"))
    text = " ".join([word for word in text.split() if word not in stop_words])

    return text
```

3. **Data Labeling**

- Toxicity scores by perspective API
- Sentiment scores by VADER, TextBlob and Empath

Toxicity Annotations: Perspective API

[17]:

```
# The Perspective API keys
PERSPECTIVE_API_KEYS = [
    "AIzaSyAMpL8JpwPU4c1nEGKCiBAiGp979r6o4-4",  # perspective-api-414709
    "AIzaSyD_-Oiitvk4OL5zgvX90Nn5TcoA23TrMlM",  # perspective-api-414723
    "AIzaSyCLQ0SAdw0-xKDEqGyTcBPO7yApPF2M3R0",  # perspe-414800
    "AIzaSyDTzo_CBwQ_5zVDojWSBMnH1jI_F6rEs7s",  # precise-antenna-414801
    "AIzaSyAt70Atcrnx2bfvFuPTwtvOV8Nf2PBPx4A",  # sound-datum-414801
    "AIzaSyBgO09nuuysiO7YNqexVZiskWhJPSv5t3A",  # perspective-api-414710
    "AIzaSyBFU4rFCLaCAVuQ0i4K3QhF_f9wBV4gBm4",  # perspective-api-414800
    "AIzaSyC8kMo6iX7iXX_lj8gx8IM0LuNS8p94UA4",  # shaped-canyon-414800
    "AIzaSyAhRHCYoYkRkQkco4NzhNuKT7Zm92BKOS8",  # perspective-api-414801
    "AIzaSyCr_b9CLWmy9Rt0f0ME74ZZmh3uT6gAwpk"  # hardy-order-414801
]

def build_client(api_key):
    """
    Build a client for a given Perspective API key.
    """
    # Create a client object
    # Reference: https://developers.google.com/codelabs/setup-perspective-api#4
    client = discovery.build(
        "commentanalyzer",  # Name
        "vlalpha1",  # Version
        developerKey=api_key,
        discoveryServiceUrl="https://commentanalyzer.googleapis.com/$discovery/rest?version=v1alpha1",
        static_discovery=False
    )
    return client

# Pre-build a client for each API key
clients = {key: build_client(key) for key in PERSPECTIVE_API_KEYS}

# Set up the iterator
api_key_iterator = itertools.cycle(PERSPECTIVE_API_KEYS)
```

[20]:

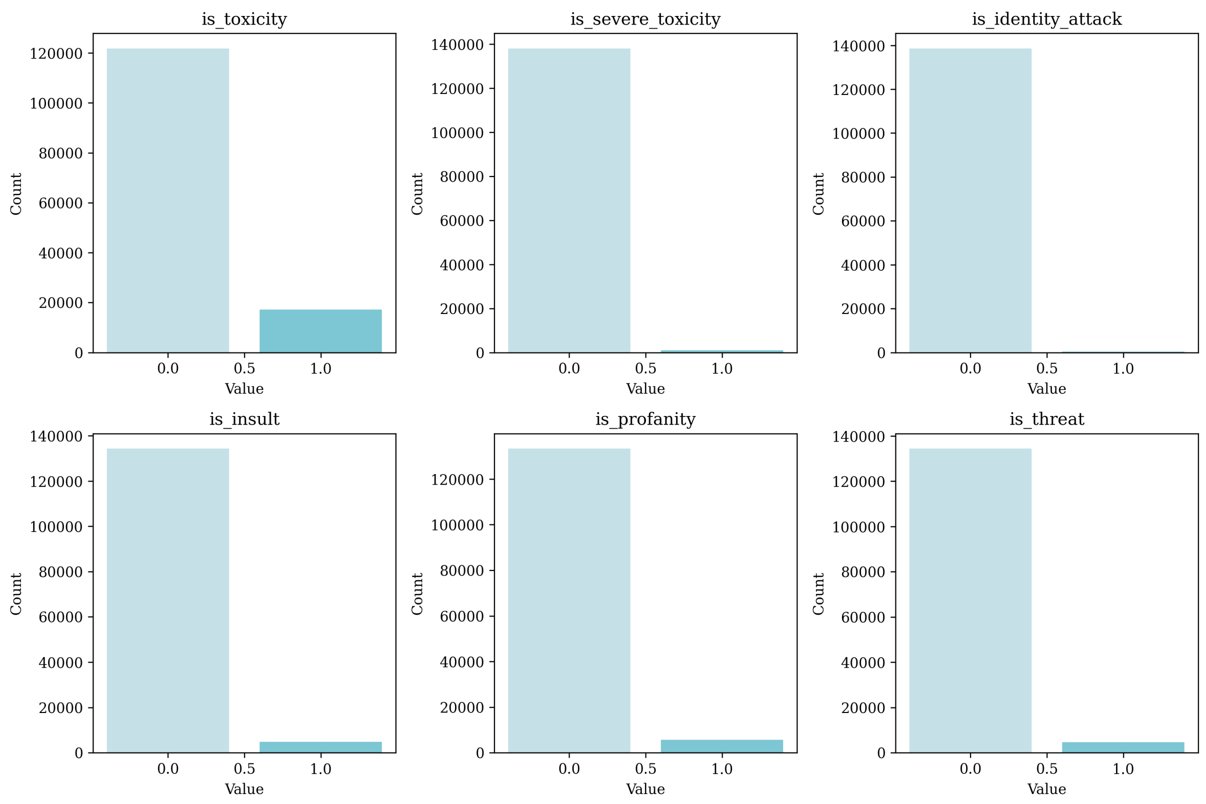
```
# %%timeit -r 1 -n 3
# Start timing
start_time = time.time()

# Compute Perspective API toxicity scores for each comment
toxicity_scores = perspective_toxicity(comments)

# End timing
print(f"Runtime: {time.time() - start_time:.4f}")
toxicity_scores.head()
```

```
Runtime: 14948.2070
```

[20]:

	toxicity	severe_toxicity	identity_attack	insult	profanity	threat
0	0.642621	0.169603	0.044097	0.342037	0.600193	0.138155
1	0.093515	0.004025	0.012943	0.023351	0.038906	0.009515
2	0.201028	0.011749	0.016059	0.025929	0.098687	0.106963
3	0.137353	0.007057	0.013345	0.028061	0.060951	0.013217
4	0.509388	0.120196	0.034301	0.249039	0.498944	0.014566

Distribution of Toxic Comments

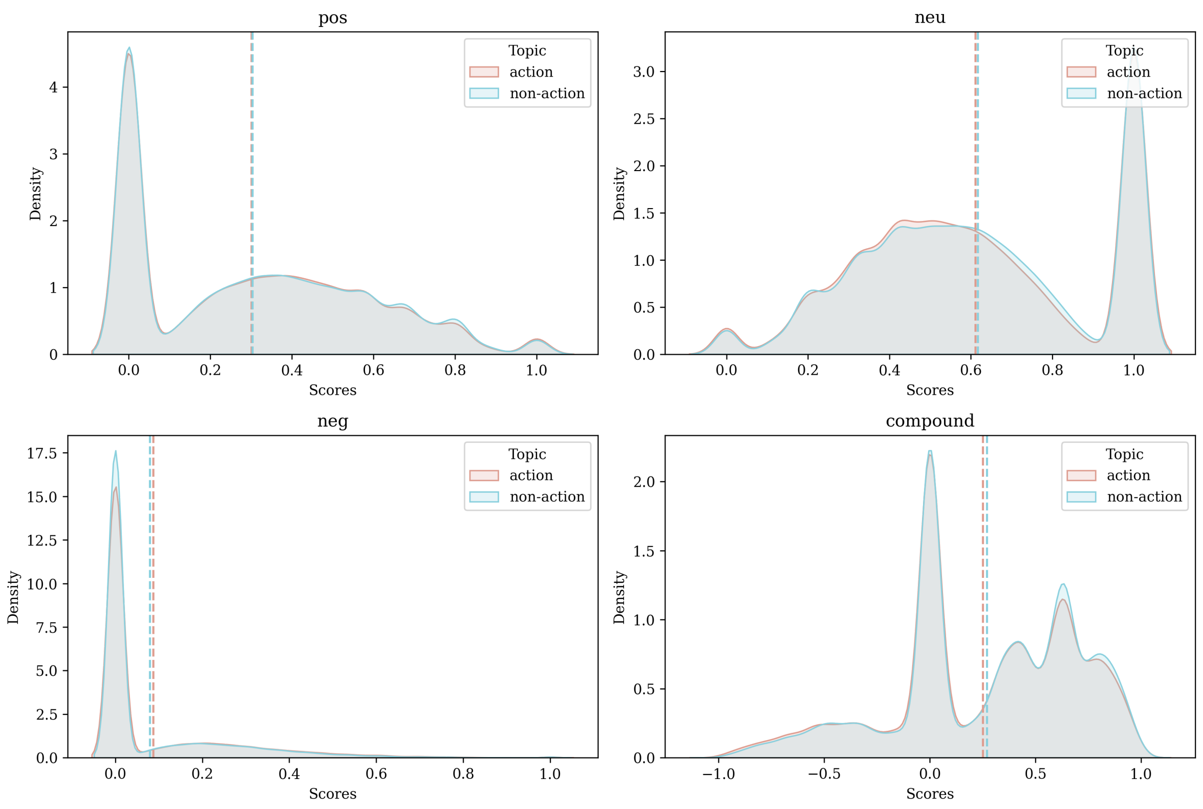


The graph above is showing the percentage of comments that are tagged with the six dimensions of Perspective API;

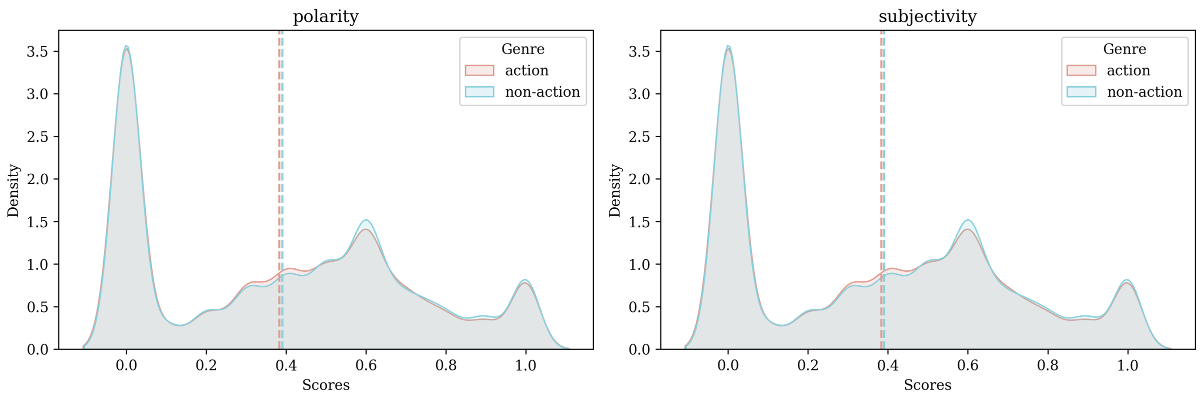
- is_toxicity (> 0.3) identified around 13% of all data and it is the largest bucket identified.
- is_identity_attack (> 0.3) identify the least number of comments among the six identifiers.

Sentiment Scoring: VADER/TextBlob/Empath

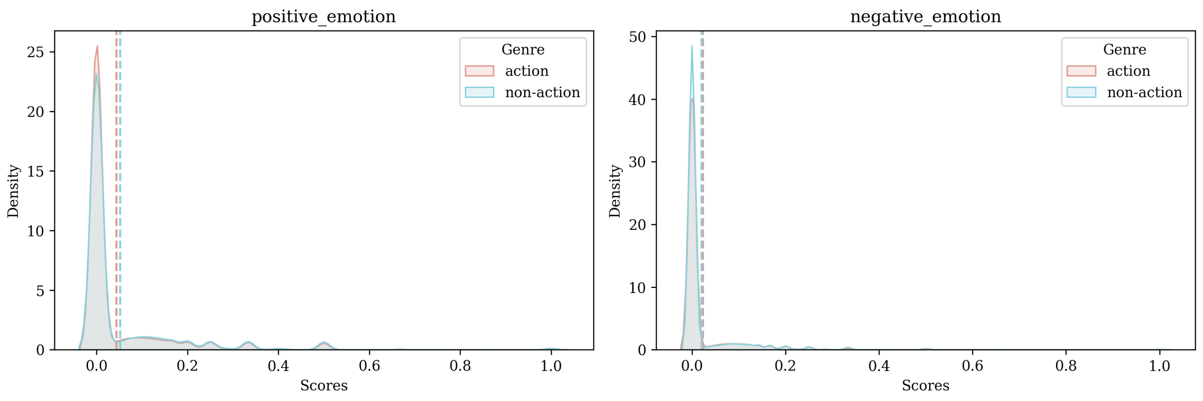
VADER Sentiment Scores by Genre



TextBlob Sentiment Scores by Genre



Empath Sentiment Scores by Genre



The above three graphs measure the sentiment scores measure by Vader, TextBlob and Empath respectively.

- We can observe that the overall distribution of action and non-action videos' sentiment scores are extremely close, amost overlapped;
- however, the vertical dotted lines, which are the mean of either action or non-action sentiments scores, are also quite close to each other.
- From Vader, we can say that action videos are showing more sentiments, positive and negative alike, and less neutrality.

Preliminary Analysis

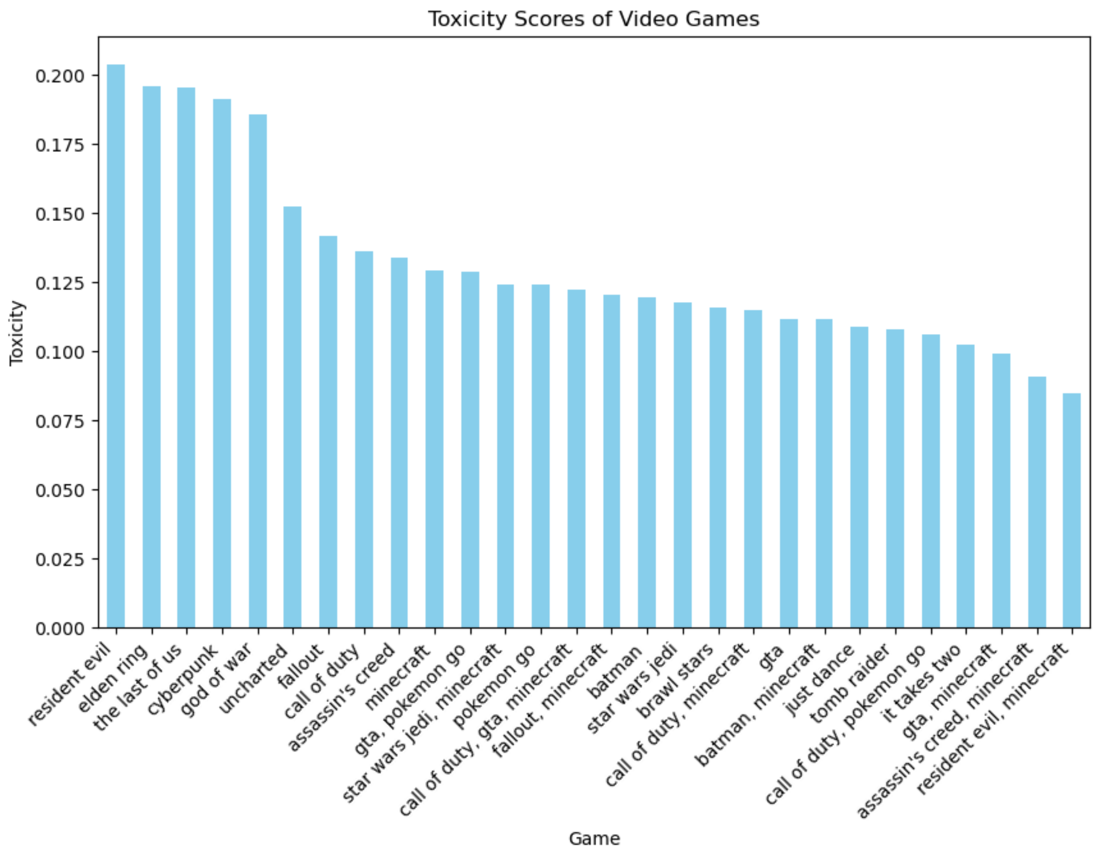
Toxicity by Game

Toxicity by Game

```
[10]: game_toxicity = simple_df.groupby('game')[sentiment_measures].mean().sort_values(by = 'toxicity', ascending=False)
```

```
[11]: game_toxicity_plot = game_toxicity['toxicity']

      plt.figure(figsize = (10, 6))
      ax = game_toxicity_plot.plot(kind='bar', color='skyblue')
      # Adding labels and title
      plt.xlabel('Game')
      ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
      plt.ylabel('Toxicity')
      plt.title('Toxicity Scores of Video Games')
```

```
[11]: Text(0.5, 1.0, 'Toxicity Scores of Video Games')
```



In the bar chart above:

- there is a pattern that action games tend to squeezed on the left hand side, which indicates higher Toxicity score.
- while non- action games like minecraft, tend to squeeze on the lower Toxicity scores side
- P.S You may see many game names are a combination of several games like 'call of duty, minecraft', this is becuase that minecraft allow player to use the gaming skins from other games.
- In other cases where multiple games are capture together, the reason might be that it is a video about both games but these situations are rare overall.

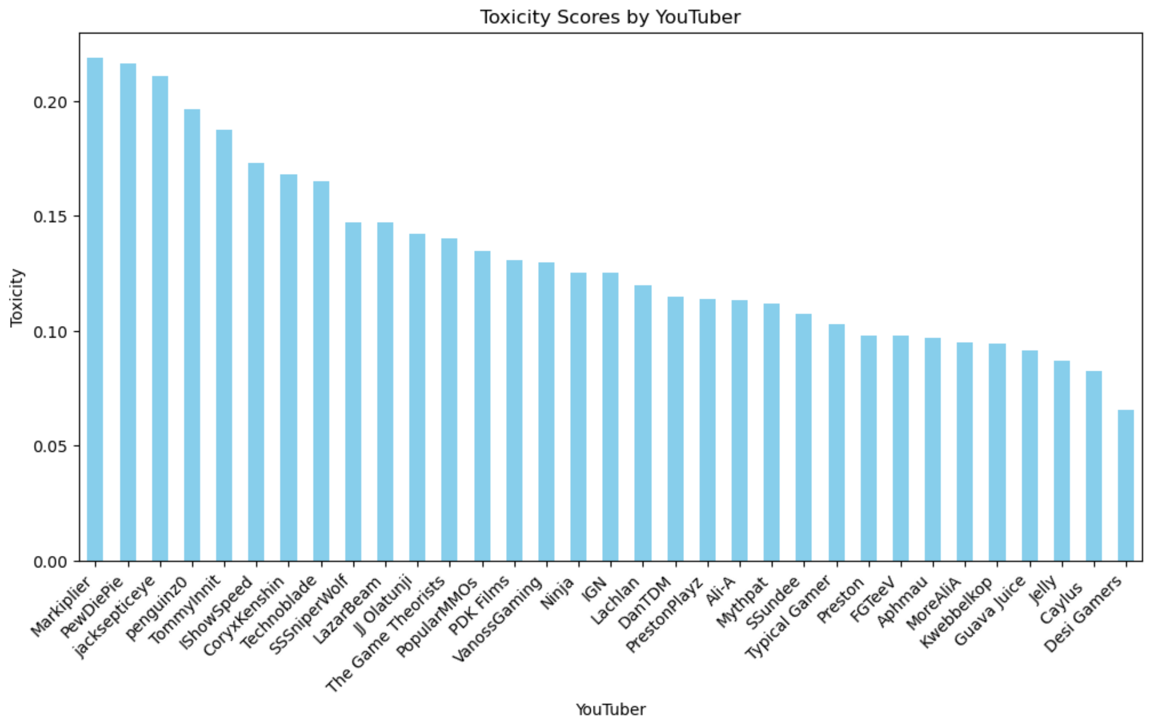
Toxicity by Channel

Toxicity by Channel

```
[13]: channel_toxicity = simple_df.groupby('channel_name')[sentiment_measures].mean().sort_values(by = 'toxicity', ascending=False)
```

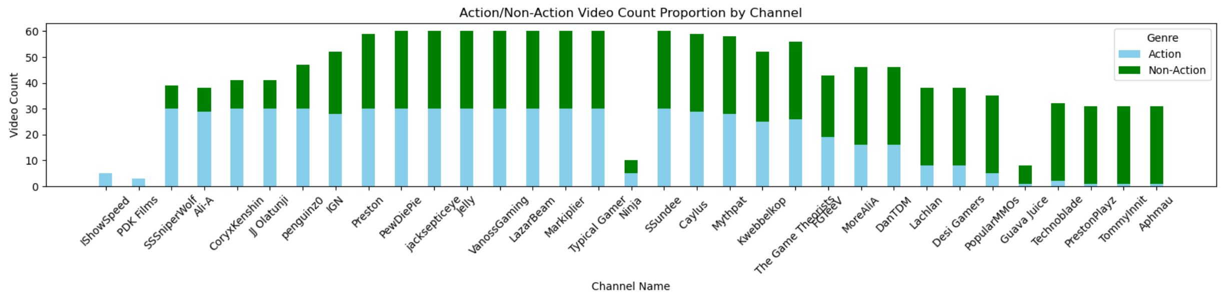
```
[14]: channel_toxicity_plot = channel_toxicity['toxicity']

      plt.figure(figsize = (12, 6))
      ax = channel_toxicity_plot.plot(kind='bar', color='skyblue')
      # Adding labels and title
      plt.xlabel('YouTuber')
      ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
      plt.ylabel('Toxicity')
      plt.title('Toxicity Scores by YouTuber')
```

```
[14]: Text(0.5, 1.0, 'Toxicity Scores by YouTuber')
```



In the bar graph above, we observe an obvious difference in the mean toxicity scores of different YouTuber.

To Investigate if the rootcause is that the higher proportion of action videos might lead to higher toxicity score we plotted the charts below.

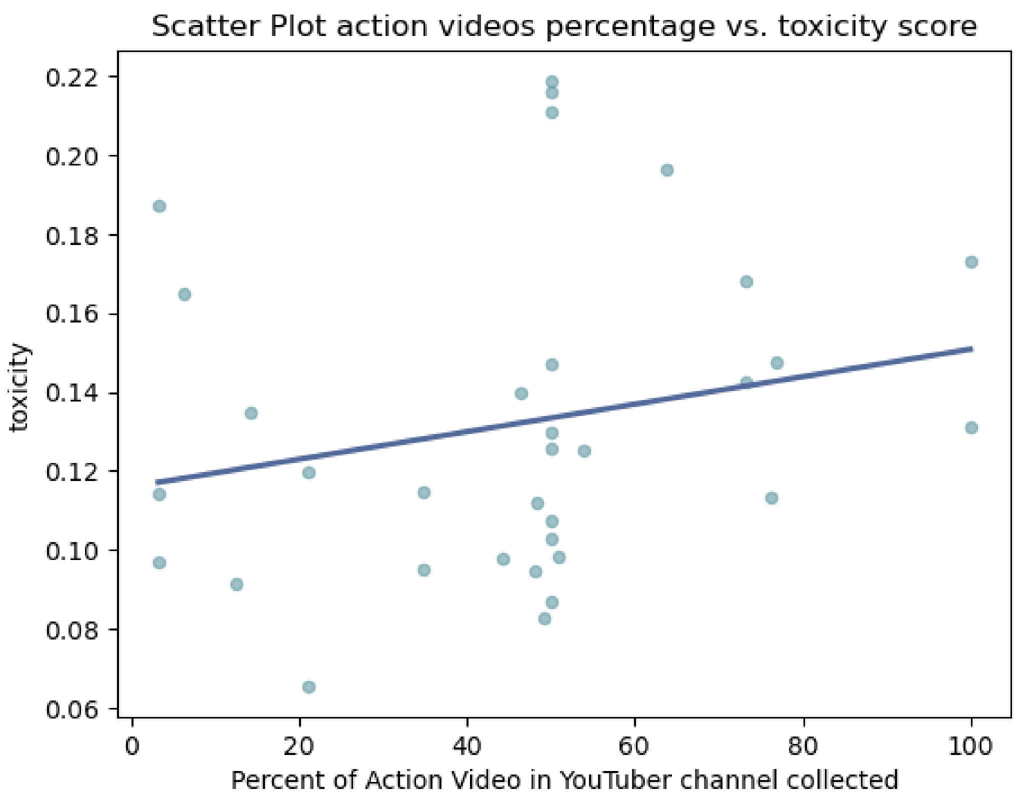


For some players that play mostly non-action games, it seems like they tend to have lower toxicity scores.

Below is a more detailed breakdown.

[21]:

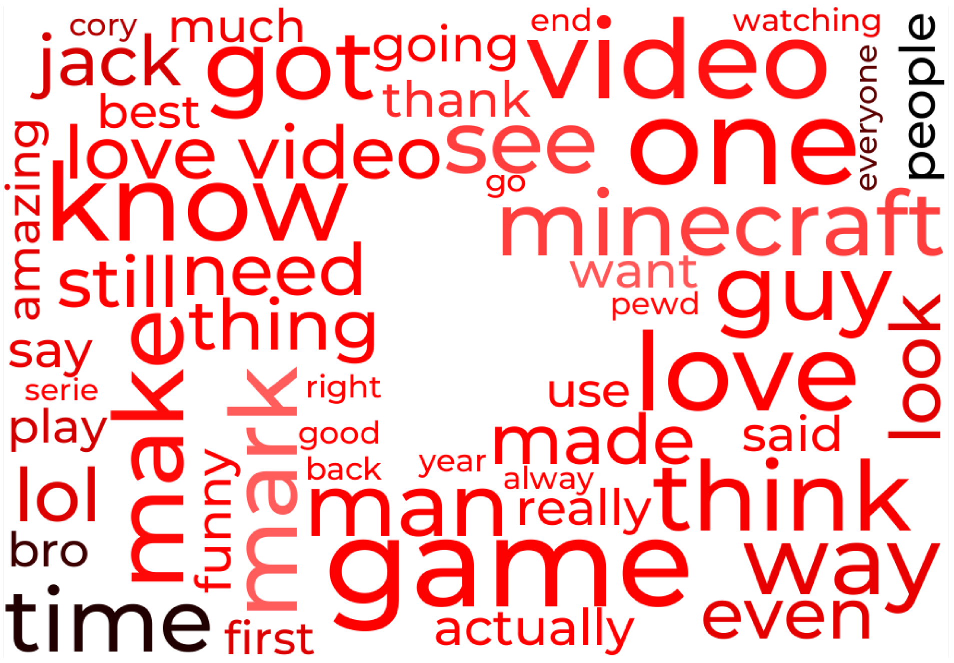
	channel_name	action_count	non_action_count	action_video_%	toxicity
0	IShowSpeed	5	0	100.00	0.173149
1	PDK Films	3	0	100.00	0.131037
2	SSSniperWolf	30	9	76.92	0.147461
3	Ali-A	29	9	76.32	0.113281
4	CoryxKenshin	30	11	73.17	0.168041
5	JJ Olatunji	30	11	73.17	0.142374
6	penguinz0	30	17	63.83	0.196428
7	IGN	28	24	53.85	0.125393
8	Preston	30	29	50.85	0.098132
9	PewDiePie	30	30	50.00	0.216254
10	jacksepticeye	30	30	50.00	0.210827
11	Jelly	30	30	50.00	0.086814
12	VanossGaming	30	30	50.00	0.129787
13	LazarBeam	30	30	50.00	0.147317
14	Markiplier	30	30	50.00	0.218835
15	Typical Gamer	30	30	50.00	0.102998
16	Ninja	5	5	50.00	0.125484
17	SSundee	30	30	50.00	0.107337
18	Caylus	29	30	49.15	0.082752
19	Mythpat	28	30	48.28	0.112104
20	Kwebbelkop	25	27	48.08	0.094685
21	The Game Theorists	26	30	46.43	0.140013
22	FGTeeV	19	24	44.19	0.097934
23	MoreAliA	16	30	34.78	0.094854
24	DanTDM	16	30	34.78	0.114798
25	Lachlan	8	30	21.05	0.119786
26	Desi Gamers	8	30	21.05	0.065423
27	PopularMMOs	5	30	14.29	0.134893
28	Guava Juice	1	7	12.50	0.091561
29	Technoblade	2	30	6.25	0.164925
30	PrestonPlayz	1	30	3.23	0.114104
31	TommyInnit	1	30	3.23	0.187283
32	Aphmau	1	30	3.23	0.096814



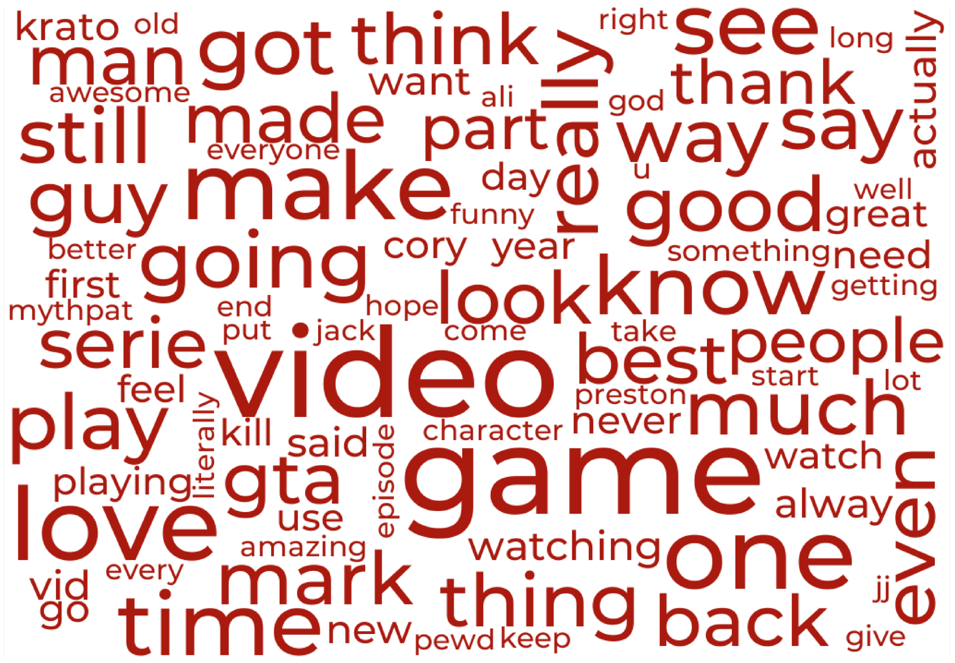
We plotted a scatter plot with trend line above. We can observe a positive relationship between Percent of Action Video in Youtuber's channel and the mean toxicity scores

- however, there are a few exception like 'Tommyinit', this youtuber mainly post non-action games videos.
- But his toxicity score is pretty high. We assume the underlying reasons might be his content, his characteristics, or his language habit;
- We plan to further investigate in the future by aalyzing the transcripts so that we can make relationships between video contents and comments section atmosphere.

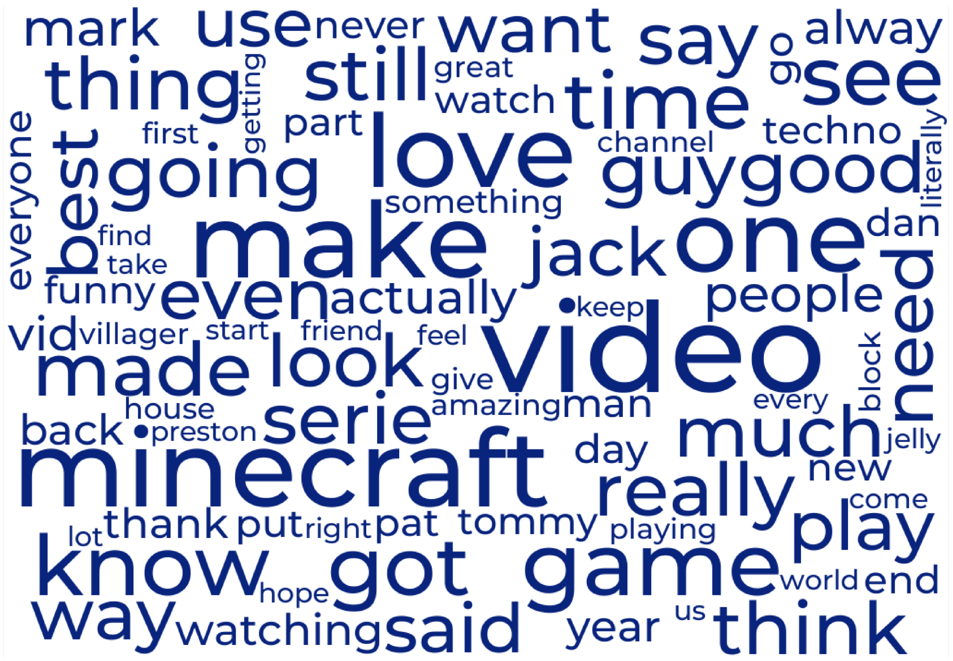
Word Clouds: All Comments



Word Cloud: Action Game Comments



Word Cloud: Non-Action Game Comments



In these 3 word clouds we do not observe eye-catching difference. It indicates the language use in action and non-action game is not very different.

Qualitative

Answer, update question/claim, summary, re-contextualization, story, relate to domain knowledge

Answer:
From our preliminary research, the action games tend to have a little higher toxicity, while may not be a reliable determinent and the influence is not strong. Also the sentiments of action and non-action games do not have obvious difference, but action games tend to arouse stronger emotions.

Updated Question:
What are other primary factors or predictors of online gaming toxicity? What are the factors influence the overall sentiment and atmosphere of YouTube comment section? YouTuber's content, topic, characteristic or language habit?

Relate to Domain Knowledge:
This finding challenges the common assumption that game category is a strong factors which influences toxicity levels and sentiment. It prompts a deeper exploration into the intricate factors contributing to online gaming toxicity. One possible reason is the gaming observer is a different group from game

players. Another reason is the game culture in YouTube gaming community is overall positive and kind, which makes the comments fairly positive.

Uncertainty, limitations, caveats

Uncertainty and limitations:

- We are not sure the fairness of our data although we fetch quite large amount of comments(130000+) for analysis.
- Our dataset is derived exclusively from top YouTube gaming channels, potentially introducing bias and limiting the generalizability of our findings to broader gaming communities.
- The predefined keyword list used for video selection might not encompass the entire spectrum of gaming content, omitting potential influential videos.

Caveats:

- Our study focuses on English-language comments, limiting applicability to non-English gaming communities.
- The Perspective API's predefined toxicity thresholds may not fully align with subjective perceptions of offensive content in gaming community.

New problems, next steps

New problems:

We observed a significant imbalance in the like-counts among comments under the same video. Certain comments received exceptionally high like-counts, what are the factors driving such popular comments?

Considering that YouTube employs its own comment filtering mechanism, the prevalence of 'toxic' comments may be limited. Can we observe obvious pattern if we analyse other emotion dimension such as disappointment, discouragement or unsatisfication?

We clearly observed different atmosphere under different youtube channel/videos. What drives such difference?

Next steps:

- Explore other factors that might influence toxicity levels: content, language habit, YouTuber characteristic and topic.
  - Collect video transcripts for video content analysis, examining the relationship between content and toxicity, as well as the relationship between language use and toxicity.
  - Collect video transcripts for YouTuber language use analysis, investigating the relationship between language use and toxicity.
  - Define different gaming Youtuber characteristics, such as serious, humourous, educational, interactive and emthusiastic, and explore the relationship between different YouTubers' characteristic and toxicity.

- Define different video topics, such as game reviews, game walkthroughs and guidance, game comparisons (such as 'I quit A for B'), game first impressions/reactions, etc., and explore the relationship between different gaming topics and toxicity.

- Explore the factors influencing engagement in gaming videos and comments.
  - Collect information about the popularity of videos/comments, such as like-count and reply-count.
  - Analyze engagement data to gain insights into which types of gaming videos are more likely to go viral, as well as comments.

Credit listing:

Chesie Yu

- Buildng code structure
- Writing code/documentations for data collection, data cleaning, text processing, toxicity and sentiment labeling, visualization and preliminary analysis

Hongfan Lu

- Writing code for data collection, processing, preliminary analysis and visualization
- Collecting gaming YouTuber channel information
- Generating observations and ideas for future exploration

Bella Wei

- Conducting Data collection and analysis
- Writing the ProjectPhase1-Summary-report

Collaboration

- Idea Generation
- Literature Review
- Reseach Question Formation
- Workflow Design
- Data collection
- Data collection result Interpretaion and Discussion
- Future direction exploration