Game-MUG: Multimodal Oriented Game Situation Understanding and Commentary Generation Dataset

Anonymous ACL submission

Abstract

The dynamic nature of esports makes the situation relatively complicated for average viewers. Esports broadcasting involves game expert casters, but the caster-dependent game commentary is not enough to fully understand the game situation. It will be richer by including diverse multimodal esports information, including audiences' talks/emotions, game audio, and game match event information. This paper introduces GAME-MUG, a new multimodal game situation understanding and audience-engaged commentary generation dataset and its strong baseline. Our dataset is collected from 2020-2022 LOL game live streams from YouTube and Twitch, and includes multimodal esports game information, including text, audio, and time-series event logs, for detecting the game situation. In addition, we also propose a new audience conversation augmented commentary dataset by covering the game situation and audience conversation understanding, and introducing a robust joint multimodal dual learning model as a baseline. We examine the model's game situation/event understanding ability and commentary generation capability to show the effectiveness of the multimodal aspects coverage and the joint integration learning approach.

1 Introduction

The recent advent of esports has led to a trendy and rapidly growing industry, capturing the attention of a large and continuously expanding global audience. Within a few seconds of a game event, numerous aspects demand attention, such as player action, skills demonstrations, team cooperation, gain and loss, and the key items contributing to the specific game events. This requires the audience to quickly digest complicated information whenever something significant happens in the game. Unlike conventional sports broadcasting like NBA games (Yu et al., 2018), where the fundamental sport's concepts are easily comprehensible, this dynamic nature of esports introduces complexity, making it challenging for the average audience to grasp the game situation fully. Therefore, we need to find a way to assist the audience in understanding the game situation better. Esports competition organisers address this issue by involving one or two casters to explain the game situation during live streaming. However, this heavily relies on the specific casters, making it difficult for them to provide more diverse information, including audience opinions, feelings, and detailed game match information. In addition, different casters may prioritise various game aspects, leaving many online esports game resources unexplained. Therefore, it is essential to explore methods for automatically generating game-related commentary that comprehensively understand the game situation, incorporating multiple aspects, such as audience discussion, emotions, and domain-specific information.

Existing esports game commentary datasets (Tanaka and Simo-Serra, 2021; Wang and Yoshinaga, 2022; Zhang et al., 2022) only utilise single-modal information as input to generate textual commentary, disregarding the potential richness of multiple aspects that can provide valuable information about the game. The lack of multimodal resources hinders researchers interested in commentary generation for Multiplayer Online Battle Arena (MOBA) games from determining the best approach to leverage information from various sources to address the game commentary task. Moreover, previous works primarily focus on providing accurate game-related facts (Wang and Yoshinaga, 2022; Zhang et al., 2022) in the generated commentary for the audience, neglecting the importance of infusing human-like qualities and emotions to engage the audience better. Due to the lack of resources, existing game commentary generation models (Tanaka and Simo-Serra, 2021; Zhang et al., 2022; Wang and Yoshinaga, 2022) simply employ an encoder-decoder to process raw game information and generate human-like commentary without fully understanding the game situations.

We introduce GAME-MUG, a multimodal game situation understanding and commentary generation dataset, and its strong baseline. Our dataset incorporates publicly available League of Legends (LOL) resources with professional caster comments from popular live streaming platforms, YouTube and Twitch, with multimodal information, including game event logs, caster speech audio, and game-related natural language discussions encompassing both human casters' commentaries and audience chats and emotions. Inspired by the joint integration of natural language understanding and generation tasks, we propose a strong baseline model that employs joint integration framework to comprehend game situations from multimodal information and generate game commentary based on this understanding of game situations and emotions. To conduct the game commentary generation, we summarise the game situation and audience conversation via multi-modality sources. Our contribution can be summarised as follows:

- Introducing a multimodal game understanding and commentary generation dataset to provide a full understanding of the game situations with caster comments and diverse information, including audience conversation, caster speech audio, and game event logs.
- Proposing a joint integration framework to generate more human-like commentary with the help of game situation understanding
- Conducting extensive experiments to show the effectiveness of multimodality in game understanding and commentary generation.

2 Related Work

2.1 Game-related Datasets

Most datasets in the game domain are proposed for commentary generation across different games, such as live-streamed MOBA games (Tanaka and Simo-Serra, 2021; Wang and Yoshinaga, 2022; Zhang et al., 2022) as well as pre-recorded esports games (Ishigaki et al., 2021; Li et al., 2019; Shah et al., 2019) or traditional sports (Yu et al., 2018), while several datasets also focus on classification tasks related to scene understanding as shown in Table 1. CS-lol (Xu et al., 2023) proposed a task of viewer comment retrieval, while MOBA-LoL (Ringer et al., 2019) proposed two classification tasks on their dataset. On top of predicting game event types, they also provide multi-view to understand the game context, by indicating the streamer's emotional state. Among all the datasets proposed for game commentary generation, most datasets allow only a single modality as the input, video only, or game information only. Some datasets allow multimodal input, but it was not for MOBA games. So far, no previous work utilises audience emotion when they build datasets to generate more human-like commentary for MOBA games. Our dataset provides both audience emotion and rich multimodal input, including audio, audience chat, and game information.

2.2 Visual-Linguistic Generation

Most works in video captioning or commentary generation for games used encoder-decoder structure (Yu et al., 2018; Li et al., 2019; Shah et al., 2019; Ishigaki et al., 2021; Tanaka and Simo-Serra, 2021; Zhang et al., 2022; Wang and Yoshinaga, 2022), and some (Tanaka and Simo-Serra, 2021; Zhang et al., 2022; Wang and Yoshinaga, 2022) experimented with several types of structures like unified encoder-decoder, pretraining method, rule-based models, and hybrid models. Some works (Li et al., 2019; Wang and Yoshinaga, 2022; Zhang et al., 2022; Ishigaki et al., 2021; Yu et al., 2018) applied recurrent seq2seq models like LSTM/GRU structures for both encoding the input and decoding for commentary, some (Tanaka and Simo-Serra, 2021; Wang and Yoshinaga, 2022; Zhang et al., 2022) used transformer-based models for generating commentary. However, no model used dense interaction/fusion among different input modalities. Previous models either lack multimodal input or concatenate different modality features as one feature vector or via simple tensor operation. The semantic gap between different modalities is ignored. In addition, no work tried dual learning of understanding game scenes and generating commentary due to limited information provided by datasets. Our method uses the audience's chats and opinions to understand the game context to facilitate the automatic generation of commentary.

3 Game-MUG

We introduce a new game commentary dataset using multimodal game situational information,

Dataset	# Matches	Modality sources	Core Task
FSN (Yu et al., 2018)	50	video, transcript	Game commentary generation
Getting Over It (Li et al., 2019)	8	video, audio, transcript	Game commentary generation
Minecraft (Shah et al., 2019)	3	video, transcript	Game commentary generation
MOBA LoL (Ringer et al., 2019)	-	video, audio, streamer's image	Streamer emotion prediction, game event type prediction
Car Racing (Ishigaki et al., 2021)	1,389	video, game info, transcript	Game commentary generation
LoL-V2T (Tanaka and Simo-Serra, 2021)	157	video, transcript	Game commentary generation
eSports Data-to-Text (Wang and Yoshinaga, 2022)	-	game info, transcript	Game commentary generation
Dota2-Commentary (Zhang et al., 2022)	234	game info, transcript	Game commentary generation
CS-lol (Xu et al., 2023)	20	transcript, chat	Viewer comment retrieval
Game-MUG (ours)	216	audio, chat, game info, transcript	Game commentary generation, game event type prediction

Table 1: Summary of existing game datasets

called Game-MUG. It features three modalities: game match event logs, audio features derived from signal data and textual discussions, such as caster comment transcript and audience chat. It comprises 70k clips with transcripts and 3.7M audience chats collected from 45 LOL competition live streams. Each live stream has an average of 4.8 individual matches, leading to 216 game matches and 15k game events. Game matches are sourced from 3 distinct leagues between 2020 and 2022, including Tencent League of Legends Pro League, League of Legends Champions Korea and World Championships. These top-tier league matches in various regions attract many views (from 507K to 7.2M), which derives abundant audience chats in multiple languages. We collect caster commentaries and audience live chats from two different livestream platforms: Twitch, which contributes 150 matches, and YouTube, which contributes 66 matches. In addition to this, we crawl game events from the League of Legends Competitive Statistics Website[1].

3.1 Data Collection

Gaming Human Commentary Transcription. We collect human commentaries by transcribing the raw live stream files[2]. Due to the substantial size of live-stream videos, we use YT-DLP and Twitch-DL only to download their high-definition (44.1kHz) audio and utilise a speech recognition model named Whisper (Radford et al., 2022) for speech-to-text conversion. Whisper is a large supervised model that implies the encoder-decoder architecture from Transformer (Vaswani et al., 2017). We use Whisper medium English model and set the compression ratio to 1.7 without previous text conditions for speech-to-text recognition, which slightly trades off the transcript accuracy but maximises its robustness. Each transcribed text is paired with its start and end timestamps in seconds.

Audience Live Chats Collection. Audience live chats are scrapped from the live stream platforms. We employ a multiplatform software named Chat Downloader to scrap the chat content from YouTube and Twitch. Because of the multilingual nature of live chats, we use Lingua to identify different languages and apply a special label called "emo" for chat instances that only include emotes or emojis. Live stream platforms automatically filters out hateful and toxic contents and we further filter out the live chats without any content [3] and associate reminders with their respective timestamps in seconds.

Game Events Collection. Game events are collected from the League of Legends Competitive Statistics Website by a scrapper; it first finds the game-related HTML tags and extracts the contents from the selected tags. It is worth noticing that sometimes the contents of the tags can be empty, which means a minion or a non-epic monster triggers this event. Our scrapper automatically populates missing contents in the tags and links them to game timestamps, constructing complete game event instances. We categorise collected game events into the following 6 different classes in our dataset: **1) Kill:** A game character is defeated; **2) Non-Epic Monster:** A jungle monster is eliminated; **3) Tower:** A turret/inhibitor is destroyed; **4) Dragon:** A dragon is eliminated; **5) Plate:** A turret's defensive barrier is shattered; **6) Nexus:** An nexus is destroyed, leading to the end of the game.

Audio Feature Extraction. It is known that human speech tone fluctuates based on emotions (Kienast and Sendlmeier, 2000) and audio modality demonstrates a notable advantage over video in capturing emotional fluctuations (Wu et al., 2021). Therefore, we extract audio features from the caster speech audio to enrich emo-

[1] https://gol.gg/esports/home/

[2] YouTube and Twitch disable their Automatic Speech Recognition tools on game live streams

[3] We de-identified all chats by masking their usernames. Details can be seen in Appendix A.

Categories	GPT-3.5	GPT-4	Tie
Kill	25.78%	51.56%	22.66%
Tower	14.20%	59.66%	26.14%
Dragon	17.71%	66.67%	15.63%
Overall	18.75%	58.75%	22.50%

Table 2: Pairwise comparison between GPT-3.5 and GPT-4 summaries, the overall agreement coefficient (Krippendorff, 2011) is 0.64 from nine human annotators. In most cases, annotators choose GPT-4 summaries over GPT-3.5 or think they are similar.

Event	# of events	Avg per match	Percentage
Kill	5,548	25.69	36.45%
Tower	2,889	13.38	18.98%
Dragon	1,646	7.62	10.81%
Other	5,138	23.79	33.76%
Total	15,221	70.47	100%

Table 3: Distributions of the more important game events in our collected dataset, where the less important ones, **Non-Epic Monster**, **Plate** and **Nexus**, are categorised into **Other** as an initial step for analysis.

tional representation within diverse domain data. The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) (Eyben et al., 2016) is commonly used for voice research and it encompasses 18 Low-Level Descriptors, which covers features related to frequency, amplitude and spectral parameters. We utilise audiofile to convert raw audio files into audio waveforms, and then extract audio features with a sampling rate of 50Hz using openSMILE (Eyben et al., 2010), a tool commonly used for vocal emotion recognition (Doğdu et al., 2022).

3.2 Data Annotation

Game Situation Summary Annotation. Inspired by the success of Standford Alpaca (Taori et al., 2023), we make use of GPT-3.5 (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023) to condense all 70,711 human commentaries into concise summaries with emotional clues from audience chats as detailed in Appendix Algorithm 1. To ensure the annotation quality, we conduct a pairwise human evaluation between the summaries from GPT-3.5 and GPT-4. As shown in Table 2, GPT-4 excels GPT-3.5 in all three categories, indicating GPT-4's summaries are better aligned with human understanding. Therefore, we choose GPT-4's summaries as ground truth annotations in our dataset.

3.3 Data Processing

Considering each live stream can be treated as a chronological sequence comprised of game events, human commentaries and live chats, we match them via their timestamps. As game events' timestamps are reset after each match, we manually adjust them to align with live stream seconds prior to the matching process. Additionally, background music before the commencement of each live stream is also removed manually, since there is no game-related factual information to help with game situation understanding.

4 Data Analysis

Our dataset includes 70,711 transcript sentences with an average duration of 12.2 secs and 3,657,611 chat instances. 15,221 game events are collected from 216 game matches. Not all events are equally important for the human caster and audience; **Kill**, **Tower**, and **Dragon** events usually attract more interest than other events. Therefore, we categorise all other events into **Other** as an initial input processing step for our following analysis in Section 4 and experiments in Section 6.3. We present the statistics of each event category in Table 3.

4.1 Game Keyword Analysis

Different from other domains, game-related data contains numerous keywords that rarely appear in everyday conversations. We manually extract 2,003 unique keywords from the caster speech transcript in our dataset and clean the typos and misspells while retaining essential abbreviations, such as character's skills denoted by Q, W, E, and R. As shown in Figure 1, extracted keywords can be categorised into 5 different classes, including skill, player, team, character and item. To better address the importance of each keyword, we compute their Term Frequency - Inverse Document Frequency (TF-IDF) based on the game events with different time windows, specifically 15 seconds and 30 seconds. The transcripts encompassed within these windows are treated as a singular document to compute TF-IDF values. This allows us to identify key terms closely associated with game events. Depending on the precise timing of the event, such a window might encapsulate one or several transcripts. This calculation is performed using the Scikit-learn library (Pedregosa et al., 2011) with normalisation. Figure 1 shows a sample visualisation of the keywords' characteristics when the

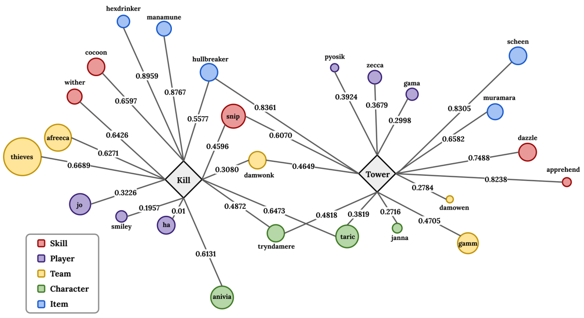

Figure 1: The visualisation for keyword analysis with top 15 words from **Kill** and **Tower** Event, where the time window is 30s. Entities related to each event, Kill and Tower, are remarkably different, such as skill 'cocoon' for **Kill** and 'apprehend' for **Tower**.

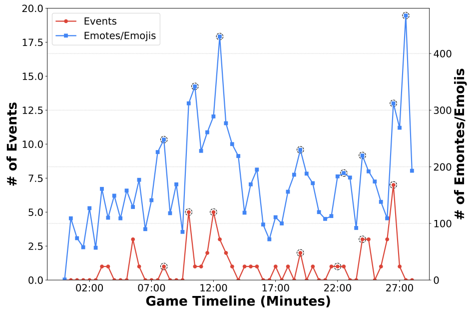

Figure 2: The concurrent plot for audience chat analysis with the numbers of emotes, emojis, and game events along the same timeline. A positive correlation can be observed between the number of audience chat emojis and the number of game events happening within the same time window.

window of time equals 30 seconds. We select the top 15 keywords for **Kill** and **Tower** events and differentiate their types by distinct colours. The size of each keyword's node depends on the normalised occurrence of the keyword, whereas the distance between the event and keyword nodes is determined by the normalised TF-IDF values. From Figure 1 we can see that **Kill** and **Tower** are more related to items to attack, skills that either increase the damage for attacking enemies or limit the ability of enemies moving to avoid damage or fighting back. This reflects the typical player's actions in games, which often involve attacking opponents, indicating that the text in our dataset effectively describes the game scene and offers a robust understanding of the situation. Moreover, we can see that team, players, and character names are frequently mentioned or discussed by commentators when these cases happened; though the names might depend on specific games, it demonstrates the multiple aspects people could focus on about the game situation.

4.2 Audience Chat Analysis

The audience tends to send many emotes and emojis in chat to express their sentiments. We retrieve emotes and emojis based on their distinct formats found in publicly available sources[4][5] and then count the number of emotes and emojis per 30-second window in each match. The counts of emotes, emojis, and game events are plotted concurrently on the same timeline, shown in Figure 2. It is not hard to discover that the number of emotes correlates with the game situation, since audiences tend to send more emotional expressions in chats to share their feelings when a dramatic turning point or a series of events happens.

[4] https://www.frankerfacez.com/emoticons/

[5] https://github.com/carpedm20/emoji/

5 Proposed Baseline

Based on Game-MUG, we proposed a joint integration framework that generates summaries of the game commentaries based on understanding the game situation through multimodal data. For game situation understanding, we implemented and fine-tuned a multimodal transformer encoder that encodes text and audio data. For game commentary generation, we employ a pre-trained decoder and encoded game information to generate new summaries. The quality of generated summaries is evaluated by both automatic metrics and humans. We partition our dataset into 206 matches for training and 10 matches for testing.

5.1 Input Processing

Given an i-th event E_i happening at t_{ei} of a game, we try to predict its event type via the multimodal information provided in our dataset and the game situation understanding module, and generate a commentary summary via the game commentary summarisation module. Taking m most recent game events which happened before E_i as a historical reference, we extract the time-series event sequence as $\mathbb{E} = \{E_{i-m}, \ldots, E_{i-2}, E_{i-1}\}$. Assuming that the input window size for transcript and chat is w, we extract a time-series sequence consisting of x transcript clips $\mathbb{T} = \{T_{s-x}, \ldots, T_{s-1}, T_s\}$, where T_s refers to the s-th transcript clip in the current game. These clips fully cover the time period from $(t_{ei} - w)$ to t_{ei}, meaning that the timestamp $(t_{ei} - w)$ falls within the time frame covered by

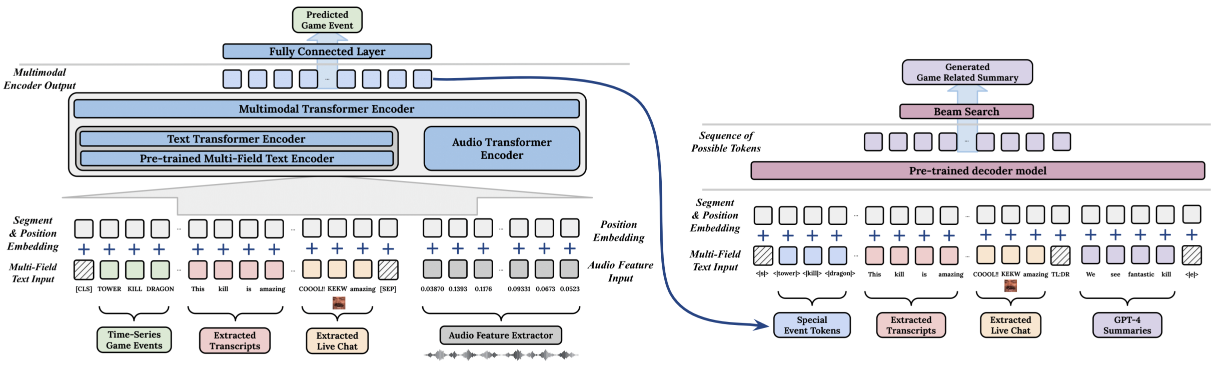

Figure 3: Joint integration framework of Game Situation Understanding and Game Commentary Summarisation

T_{s-x}, and t_{ei} falls within the time frame covered by T_s. The time-series sequence of chats $\mathbb{C}$ is extracted based on their specific timestamps between $(t_{ei} - w)$ and t_{ei}. For the audio component, given the window size w_a, the audio feature sequence is extracted as $\mathbb{A}$ within the time period between $(t_{ei} - w_a)$ and t_{ei}. This results in a vector consisting of $w_a * 50$ values that serve as the input for the audio transformer, given that the audio features are sampled at a rate of 50Hz.

5.2 Game Situation Understanding

The model architecture is shown on the left of Figure 3. On the text side, the input is a combination of multi-field sequential time-series data from previous event $\mathbb{E}$, caster transcript $\mathbb{T}$ and audience chat $\mathbb{C}$, with graphical emotional expressions in chats being converted into their text representation. Since chats tend to contain many repetitions in phrases and emotions, we truncate the input sequence up to 256 tokens. Following the approaches in BERT (Devlin et al., 2019), we insert a [CLS] token at the beginning and a [SEP] token at the end of the input sequence, creating the input embeddings by summing the token, segment, and position embeddings. These input embeddings are initially passed into a pre-trained multi-field text encoder. The [CLS] token output from this pre-trained multi-field text encoder is then forwarded to the text transformer encoder to project the text representation into a common space. On the audio side, the combination of audio feature $\mathbb{A}$ and position embedding are fed into an audio transformer, which maps the audio into the same common space as the text. The text and audio representations are then concatenated to form a single vector, which serves as the input for the multimodal transformer encoder followed by a fully connected layer to predict the subsequent game event. We take advantage of existing pre-trained models in our multi-field text encoder including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021), and XLNet (Yang et al., 2019). More details can be found in Section 6.1.

5.3 Game Commentary Summarisation

After fine-tuning the game situation understanding model, we obtain the event representations from it before the fully connected layer and incorporate these representations along with transcripts and chats into the pre-trained generative model for summarisation. We calculate the mean of each event representation by inference the trained game situation understanding model with all the matches in our dataset to get the special event embeddings. These embeddings are then added to the decoder models' vocabulary as <|kill|>, <|tower|> and <|dragon|> to enhance efficiency during summary generation. Similar to the encoder model, we truncate the chat sequence up to 256 tokens for emotion extraction before combination them with special event tokens and transcripts. As shown in the right of Figure 3, a special [TL;DR] token and GPT-4 summary are concatenated to the sequence as a reference during fine-tuning. We utilise two different pre-trained decoders, including GPT-2 (Radford et al., 2019) and Pythia (Biderman et al., 2023). More details can be found in Section 6.1.

6 Experiments and Results

6.1 Experiment Setup

Game Situation Understanding We test four pre-trained encoder models with their large settings as the baseline multi-field text encoders: BERTLARGE, RoBERTaLARGE, DeBERTaV3LARGE, and XLNetLARGE. The text and au-

Chat	Audio	Game Events	BERT				DeBERTaV3				RoBERTa				XLNet			
			Kill	Tower	Dragon	All	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All
✘	✘	✘	77.98	47.75	8.45	61.97	79.46	62.16	1.41	65.06	79.17	62.16	8.45	65.83	93.75	10.81	4.23	63.71
✔	✘	✘	86.01	20.72	9.86	61.58	81.55	62.16	0.00	66.22	79.46	59.46	7.04	65.25	96.43	0.90	5.63	63.51
✘	✔	✘	83.63	37.84	14.08	64.29	77.08	36.94	49.30	64.67	78.57	62.16	25.35	67.76	72.02	55.86	22.54	61.78
✘	✘	✔	80.55	51.35	17.19	64.96	72.35	61.26	17.19	62.18	78.50	58.56	35.94	67.95	95.22	15.32	0.00	63.25
✔	✔	✘	75.00	48.65	11.27	60.62	82.44	43.24	43.66	68.73	77.38	63.06	15.49	65.83	67.86	53.15	14.08	57.34
✔	✘	✔	83.22	51.35	18.03	66.81	81.82	58.56	40.98	70.74	80.07	55.86	36.07	68.34	80.07	62.16	21.31	67.90
✘	✔	✔	84.97	32.43	42.62	66.59	79.72	49.55	34.43	66.38	84.62	51.35	26.23	68.78	76.22	60.36	21.31	65.07
✔	✔	✔	83.57	43.24	18.03	65.07	86.71	31.53	59.02	69.65	80.42	52.25	31.15	67.03	83.92	53.15	18.03	67.69

Table 4: The effect of Chat, Audio and previous Game Events on 2 different Game Situation Understanding Models.

dio transformer encoder and the multimodal transformer encoder are all 8-head and 6-layer encoder structures and 1024 embedding dimension. The entire model is trained using AdamW (Loshchilov and Hutter, 2019) with 2 epochs for each instance, with a dropout value of 0.1 (Srivastava et al., 2014), a learning rate of 1e-6, and a learning rate decay rate of 0.95 for every 2 epochs. **Game Commentary Summarisation** We adopt two pre-trained decoder models as the baseline commentary summarisation models: 762M GPT2 with 1280 dimension size and 410M Pythia with 1024 embedding size. We apply Principal Component Analysis (Wold et al., 1987) to the game event embeddings when their dimensions are larger than the embeddings of pre-trained models for fine-tuning consistency. All models are trained using AdamW for 3 epochs, with a learning rate of 1e-5, and a warmup step of 5. Our implementations are based on PyTorch (Paszke et al., 2019) and HuggingFace Transformers (Wolf et al., 2020), with the help of Scikit-learn (Buitinck et al., 2013). All experiments are run on a test bench with 24GB NVIDIA RTX 3090 GPU.

6.2 Evaluation Metrics

We evaluate the game situation understanding model with a multi-class accuracy metric, directly comparing the predicted game event with the ground truth for each event class. Generated summaries are evaluated with ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020), common automatic evaluation metrics. To have the best correlation with humans, we choose a RoBERTaLARGE version of BERTScore, which deploys a RoBERTa model to compare the similarity between the model generations and references. All results are reported for a single run of the experiments.

6.3 Results

Overall Performance As illustrated in Table 4, when all input features are utilised, DeBERTaV3 notably outperforms the others in overall accuracy as well as **Kill** and **Dragon** categories by trading

Special Event Token	GPT2		Pythia	
	BertScore	ROUGE-L	BertScore	ROUGE-L
✘	76.15	18.52	74.45	13.24
✔	76.38	17.10	75.37	15.98

Table 5: The effect of special event tokens on 2 different Game Commentary Summarisation Models.

off the performance on **Tower**. Trailing behind DeBERTaV3, the overall performance of RoBERTa and XLNet is similar, with a margin difference of less than 1%. It is worth noting that RoBERTa excels in the **Dragon** category, while XLNet excels in the **Kill** and **Tower** categories. Although BERT achieves an overall accuracy of 65.07%, it ranks last among the four encoder variants. This is likely attributable to the other models' more robust optimisation built upon BERT's architecture. In addition, all models produce better prediction accuracy for **Kill** than for **Tower** and **Dragon**. This trend is primarily due to the imbalanced event data since the average number of **Kill** instances per match is 25.69, which is double the average number of **Tower** instances (11.62) and triple the average number of **Dragon** instances (7.62). Regarding the game commentary summarisation results presented in Table 5, we note that GPT2 consistently outperforms Pythia across both evaluation metrics, irrespective of special event tokens.

Ablation Studies To further analyse the effectiveness of our data, we conduct ablation studies to compare 3 different input combinations with transcripts for the game situation understanding model: **1) Audio:** with and without audio features as part of the sequence input; **2) Chat:** with and without chat as part of the sequence input; **3) Game Events:** with and without game events as part of the sequence input. The results are presented in Table 4. We observed that supplementing the model with additional input data improves its capability for understanding game situations. This results in a noticeable performance increase across all three models, particularly for the rare **Dragon** event, albeit with

Audio	BERT				DeBERTaV3			
	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All
5s	80.55	42.34	25.00	63.89	83.62	35.14	57.81	68.59
10s	83.62	37.84	23.44	64.53	83.28	35.14	59.38	68.59
15s	84.30	40.54	18.75	**64.96**	86.35	28.83	57.81	**68.80**

Table 6: Hyperparameter testing on the Game Situation Understanding Models for different audio time windows (rounded to the nearest integer in order to obtain enough data to match the audio transformer embedding size which should be a multiple of 8), where input transcript and chat time windows are 30s, and the number of previous game events is 5. A larger audio time window may lead to higher performance with a small margin.

Game Events	BERT				DeBERTaV3			
	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All
3	85.39	29.73	18.84	63.32	88.64	25.23	53.62	69.26
5	84.30	40.54	18.75	**64.96**	86.35	28.83	57.81	68.80
7	80.58	40.91	21.67	62.95	85.25	42.73	56.67	**70.98**
9	79.32	50.00	12.50	63.32	71.80	64.15	0.00	60.51

Table 7: Hyperparameter testing on the Game Situation Understanding Model for different numbers of previous game events, where input transcript and chat time windows are 30s and the audio time window is 15s. A large number of previous game events may include less relevant histories and lead to a worse performance.

Category	GPT2			Pythia		
	Event	Coherence	Overall	Event	Coherence	Overall
Kill	75.31%	75.31%	66.67%	24.69%	24.69%	33.33%
Tower	60.74%	59.26%	59.26%	39.26%	40.74%	40.74%
Dragon	61.62%	66.67%	59.60%	38.38%	33.33%	40.40%
All	64.76%	65.71%	61.27%	35.24%	34.29%	38.73%

Table 8: Human evaluation comparison between GPT2 and Pythia summaries. Appendix D and Figure 4 show the details for event inclusion, coherence and overall quality. GPT2 gains better support from human annotators across all 3 aspects compared to Pythia.

a slight trade-off in performance for other events. Specifically, individually incorporating audio or previous game events into the transcript yields a greater improvement than adding chat data alone. Furthermore, combining two types of additional inputs surpasses the performance achieved with just a single extra input. We also conduct experiments both in the presence and absence of the **Special Event Token**, defined as the intermediate embedding before the fully connected layer within the game situation understanding model, as illustrated in Figure 3. Other inputs, such as transcripts, chats, and GPT-4 summaries, are essential for fine-tuning since omitting any of these causes a significant drop in generation performance. The results of these experiments are shown in Table 5. We observed the addition of a special event token can guide model generation, leading to improvements in BertScore for both GPT2 and Pythia.

Hyperparameter Testing The audio hyperparameter testing for the three different variations of the Game Situation Understanding Model is in Table 6, where input transcript and chat time windows are set to 30 seconds, and the number of previous game events are set to 5. We observe that the performance of each model is barely influenced by the input length of the audio features, as the difference is within a 1% margin. We also explore the effectiveness of different numbers of previous game events and results are shown in Table 7, where input transcript and chat time windows are set to 30 seconds, and the audio time window is set to 15 seconds. Increasing the number of previous game events improves the models' aggregate performance up until a specific threshold. However, it is observed that when this threshold is surpassed, there is a discernible decrement in performance. We hypothesise that the performance decline is due to the extended length of the previous events, which have less correlation with the target event.

Human Evaluation Automatic metrics may not correlate well with human judgments in different aspects (Durmus et al., 2020), therefore we conduct the human evaluation to enrich the comprehensiveness of the results. We randomly collected testing samples for evaluating the summaries from GPT2 and Pythia and recruited nine workers, all with general background knowledge of League of Legends for evaluation, resulting in 1,890 instances of human feedback. As shown in Table 8, summarisations of GPT2 are more preferred by humans in all categories which aligns with the results from automatic evaluation metrics.

7 Conclusions

We introduce GAME-MUG, a multimodal dataset for game situation understanding and game commentary generation, and propose a joint integration baseline model. It contains diverse game-related information from game event logs, caster comments, audience conversations and caster speech audio. The combination of multimodal data improves the model's understanding of the game situations while providing the game situation information leads to more human-like game commentary generation. Finally, we will make our dataset publicly available, hoping it will lead to novel applications.

Limitations

In this work, we only consider League of Legends as the representation of the MOBA game due to its popularity (Duan et al., 2023). This constrains the range of game scenarios covered by GAME-MUG and consequently limits the scope of potential applications built upon it. We encourage future studies to incorporate a variety of MOBA games to further enrich the diversity of game situations. Furthermore, we used GPT-generated summaries for annotation in our dataset. We may apply other generative AI if new models emerge later on.

Ethics Statement

All the experiments strictly follow the Code of Ethics. In Section 6.3 human evaluation and Appendix, we include the instructions and screenshots of the interface in the human evaluation and report the background of human judges. More detailed information about the recruitment process will be shared after the paper acceptance. We inform the human evaluators what the task is about and tell them that their responses will be used to assess the ability of language generation models.

References

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cem Doğdu, Thomas Kessler, Dana Schneider, Maha Shadaydeh, and Stefan R. Schweinberger. 2022. A comparison of machine learning algorithms and feature sets for automatic vocal emotion recognition in speech. *Sensors*, 22(19).

Peng Duan, Xiaohui Wang, Allan Yijia Zhang, and Bin Ji. 2023. Case analysis on world's top e-sports events. In *Electronic Sports Industry in China: An Overview*, pages 119–133. Springer.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Florian Eyben, Klaus R. Scherer, Björn W. Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. 2016. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1459–1462, New York, NY, USA. Association for Computing Machinery.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Tatsuya Ishigaki, Goran Topic, Yumi Hamazono, Hiroshi Noji, Ichiro Kobayashi, Yusuke Miyao, and Hiroya Takamura. 2021. Generating racing game commentary from vision, language, and structured data. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 103–113, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Miriam Kienast and Walter F. Sendlmeier. 2000. Acoustical analysis of spectral and temporal changes in emotional speech. In *Proc. ITRW on Speech and Emotion*, pages 92–97.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Chengxi Li, Sagar Gandhi, and Brent Harrison. 2019. End-to-end let's play commentary generation using multi-modal video representations. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, FDG '19, New York, NY, USA. Association for Computing Machinery.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Charles Ringer, James Alfred Walker, and Mihalis A. Nicolaou. 2019. Multimodal joint emotion and game context recognition in league of legends livestreams. In *2019 IEEE Conference on Games (CoG)*, pages 1–8.

Shukan Shah, Matthew Guzdial, and Mark O Riedl. 2019. Automated let's play commentary. *arXiv preprint arXiv:1909.02195*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Tsunehiko Tanaka and Edgar Simo-Serra. 2021. Lol-v2t: Large-scale esports video description dataset. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4552–4561.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Zihan Wang and Naoki Yoshinaga. 2022. Esports data-to-commentary generation on large-scale data-to-text dataset.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingyao Wu, Ting Dang, Vidhyasaharan Sethu, and Eliathamby Ambikairajah. 2021. Multimodal affect models: An investigation of relative salience of audio and visual cues for emotion prediction. *Frontiers in Computer Science*, 3.

Junjie H. Xu, Yu Nakano, Lingrong Kong, and Kojiro Iizuka. 2023. Cs-lol: A dataset of viewer comment with scene in e-sports live-streaming. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, CHIIR '23, page 422–426, New York, NY, USA. Association for Computing Machinery.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Huanyu Yu, Shuo Cheng, Bingbing Ni, Minsi Wang, Jian Zhang, and Xiaokang Yang. 2018. Fine-grained video captioning for sports narrative. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6006–6015.

Dawei Zhang, Sixing Wu, Yao Guo, and Xiangqun Chen. 2022. MOBA-E2C: Generating MOBA game commentaries via capturing highlight events from the meta-data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4545–4556, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A De-Identification of User Chats

To ensure the anonymity and privacy of individuals involved in the live chats, we implemented a de-identification protocol. The primary objective of this protocol is to mask any information that could potentially reveal the identity of a chat participant. We directly remove all original usernames associated with the chats, ensuring it is infeasible to reverse engineer the original usernames. All de-identified chats are stored in plain text format, without any identifying information. The original raw data are permanently deleted after the de-identification process. By taking these steps, we ensure that our data collection and analysis processes align with ethical guidelines and data protection regulations.

B Prompt Design

Algorithm 1 illustrates the approach for querying the GPT-4 API. We set the background information as watching a live game streaming via a system prompt. Whenever a game event occurs, we forward the commentary and live chat content to the GPT-4 API through the summary prompts. We design several prompt parameters to guide the GPT-4 generation: <game streaming platform> indicates different live stream platforms, <number of summary words> control the number of generated words, and <game-related topics> adjusts the generated summary to focus on different aspects, such as on player, character, event or overall situation.

C Full Hyperparamemter Testing Results

The complete hyperparameter results are displayed in Table 9. We conducted experiments using 15s and 30s time windows for transcripts and chats, and 5s, 10s, and 15s time windows for audio. Additionally, we experimented with time-series events ranging from 3 to 10.

D Human Evaluation

We recruited nine volunteers aged between 25 and 30, all holding at least a Bachelor's degree, to participate in the human evaluation. The group was composed of three females and six males, each with a general understanding of League of Legends. While one participant was a native English speaker, the other eight were proficient in English. For the human evaluation survey, participants were presented with the original transcript, the truncated chat, and the generated summaries from the baseline models. They are then asked to rank the summaries based on the following four criteria:

- **Game Event Information:** The quality of summaries in terms of the game event-related expressions.
- **Coherence:** The quality of summaries in terms of fluency and logic.
- **Overall**: The overall quality of summaries regarding the above criteria and any other game-related criteria.

The sample evaluation questions are shown in Figure 4.

Algorithm 1 Game Situation Summary Annotation

Require: <game streaming platform>, <number of summary words>, <game-related topics>
Ensure: Input human transcript and audience chat

procedure BACKGROUND INFORMATION
  System Prompt: You are watching the League of Legends Competition live stream from <game streaming platform> with other audiences.
end procedure

procedure GAME SITUATION SUMMARY ANNOTATION
  Summary Prompt: Based on the <system prompt>, generate a one-sentence summary between <number of summary words> from this human transcript highlighting <game-related topics>, while incorporating the audience's emotions from this <game streaming platform> audience chat.
end procedure

Transcript + Chat	Audio	Game Events	BERT				RoBERTa				DeBERTaV3				XLNet			
			Kill	Tower	Dragon	All	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All
15s	5s	3	90.26	16.22	1.45	60.86	85.71	29.73	0.00	60.86	73.38	29.73	0.00	53.07	77.92	27.93	0.00	55.53
15s	5s	4	85.67	36.04	5.97	62.97	80.33	58.56	7.46	65.06	79.00	58.56	7.46	64.23	82.00	30.63	1.49	58.79
15s	5s	5	86.69	33.33	12.50	63.89	80.20	62.16	4.69	65.60	82.25	53.15	23.44	67.31	92.15	22.52	0.00	63.03
15s	5s	6	84.62	31.53	19.67	63.10	79.72	59.46	14.75	66.16	83.22	50.45	21.31	67.03	83.92	56.76	0.00	66.16
15s	5s	7	85.25	30.91	16.67	62.72	83.09	51.82	25.00	67.63	83.81	48.18	30.00	67.86	85.25	55.45	0.00	66.52
15s	5s	8	82.05	36.70	17.86	62.56	78.02	55.96	23.21	65.53	83.15	42.20	35.71	66.89	86.08	62.39	0.00	69.18
15s	5s	9	80.83	33.02	17.86	60.75	79.32	55.66	26.79	66.59	83.46	38.68	39.29	66.59	87.22	53.77	1.79	67.76
15s	5s	10	81.92	33.96	15.38	61.48	78.46	57.55	25.00	66.51	83.08	43.40	46.15	68.42	87.31	55.66	7.69	69.38
15s	10s	3	86.04	30.63	4.35	61.89	69.16	57.66	7.25	57.79	71.43	61.26	4.35	59.63	65.58	33.33	13.04	50.82
15s	10s	4	87.00	36.04	7.46	64.02	84.33	33.33	31.34	65.06	77.67	61.26	11.94	64.64	74.00	33.33	10.45	55.65
15s	10s	5	84.98	37.84	12.50	63.89	79.52	55.86	28.12	66.88	81.91	49.55	17.19	65.38	83.96	42.34	7.81	63.68
15s	10s	6	81.82	34.23	21.31	62.23	79.72	56.76	24.59	66.81	81.12	44.14	40.98	66.81	79.37	54.05	1.64	62.88
15s	10s	7	83.09	30.91	13.33	60.94	80.94	46.36	35.00	66.29	84.17	45.45	35.00	68.08	84.53	55.45	3.33	66.52
15s	10s	8	81.68	36.70	16.07	62.10	78.75	52.29	28.57	65.75	84.98	43.12	39.29	68.72	86.45	58.72	10.71	69.86
15s	10s	9	82.33	31.13	19.64	61.45	79.70	57.55	23.21	66.82	81.95	40.57	48.21	67.29	86.47	50.00	7.14	67.06
15s	10s	10	80.38	35.85	15.38	61.00	82.31	58.49	17.31	68.18	82.69	42.45	44.23	67.70	86.54	57.55	3.85	68.90
15s	15s	3	82.14	36.94	2.90	60.66	70.13	65.77	2.90	59.63	80.84	54.05	11.59	64.96	59.42	60.36	8.70	52.46
15s	15s	4	84.33	44.14	8.96	64.44	75.67	63.06	5.97	62.97	80.00	54.05	23.88	66.11	63.00	53.15	14.93	53.97
15s	15s	5	83.28	39.64	10.94	63.03	74.74	69.37	6.25	64.10	85.32	36.94	34.38	66.88	73.38	66.67	4.69	62.39
15s	15s	6	81.82	38.74	13.11	62.23	81.12	54.95	26.23	67.47	84.27	49.55	34.43	69.21	76.92	66.67	3.28	64.63
15s	15s	7	83.45	30.91	15.00	61.38	80.22	50.91	28.33	66.07	86.33	37.27	36.67	67.63	82.01	56.36	1.67	64.96
15s	15s	8	79.12	38.53	14.29	60.73	77.29	61.47	26.79	66.89	87.18	43.12	41.07	70.32	83.15	62.39	3.57	67.81
15s	15s	9	81.58	33.02	8.93	60.05	77.44	58.49	19.64	65.19	81.95	41.51	42.86	66.82	85.71	52.83	3.57	66.82
15s	15s	10	80.00	37.74	17.31	61.48	81.92	56.60	21.15	67.94	86.54	38.68	44.23	69.14	84.62	53.77	11.54	67.70
30s	5s	3	80.52	34.23	20.29	61.48	81.49	48.65	44.93	68.85	82.47	32.43	71.01	69.47	85.71	54.05	10.14	67.83
30s	5s	4	81.33	43.24	23.88	64.44	79.00	51.35	40.30	67.15	86.00	44.14	40.30	69.87	84.33	51.35	17.91	67.36
30s	5s	5	80.55	42.34	25.00	63.89	81.91	50.45	37.50	68.38	83.62	35.14	57.81	68.59	82.94	50.45	23.44	67.09
30s	5s	6	83.22	43.24	26.23	65.94	80.42	55.86	36.07	68.56	83.57	31.53	57.38	67.47	84.27	47.75	21.31	67.03
30s	5s	7	81.29	41.82	25.00	64.06	80.58	57.27	30.00	68.08	83.81	40.00	56.67	69.42	84.53	48.18	25.00	67.63
30s	5s	8	79.12	46.79	14.29	62.79	81.68	56.88	33.93	69.41	83.15	41.28	57.14	69.41	84.98	50.46	17.86	67.81
30s	5s	9	78.57	46.23	14.29	62.15	77.82	51.89	32.14	65.42	68.42	36.79	0.00	51.64	86.47	46.23	19.64	67.76
30s	5s	10	78.85	52.83	9.62	63.64	81.15	55.66	30.77	68.42	71.15	71.70	0.00	62.44	85.00	46.23	21.15	67.22
30s	10s	3	81.17	35.14	23.19	62.50	81.49	48.65	47.83	69.26	83.77	35.14	56.52	68.85	83.77	55.86	15.94	67.83
30s	10s	4	82.33	40.54	25.37	64.64	77.33	49.55	43.28	66.11	85.67	30.63	50.75	67.99	82.67	52.25	17.91	66.53
30s	10s	5	83.62	37.84	23.44	64.53	81.23	50.45	35.94	67.74	83.28	35.14	59.38	68.59	83.96	53.15	17.19	67.52
30s	10s	6	81.82	45.05	22.95	65.07	79.02	54.95	32.79	67.03	83.57	38.74	59.02	69.43	85.31	52.25	16.39	68.12
30s	10s	7	80.94	39.09	25.00	63.17	82.73	48.18	31.67	67.41	83.09	39.09	56.67	68.75	83.09	49.09	20.00	66.29
30s	10s	8	79.12	51.38	14.29	63.93	81.32	53.21	35.71	68.49	83.88	42.20	55.36	69.86	84.62	51.38	19.64	68.04
30s	10s	9	78.20	43.40	12.50	60.98	80.83	50.94	32.14	67.06	69.17	74.53	0.00	61.45	87.97	45.28	21.43	68.69
30s	10s	10	80.00	49.06	11.54	63.64	80.77	56.60	30.77	68.42	74.62	69.81	0.00	64.11	86.92	48.11	23.08	69.14
30s	15s	3	85.39	29.73	18.84	63.32	81.17	48.65	36.23	67.42	88.64	25.23	53.62	69.26	79.55	54.95	24.64	66.19
30s	15s	4	83.67	37.84	23.88	64.64	79.67	46.85	32.84	65.48	86.67	33.33	52.24	69.46	83.00	47.75	20.90	66.11
30s	15s	5	84.30	40.54	18.75	64.96	82.59	49.55	32.81	67.95	86.35	28.83	57.81	68.80	82.94	53.15	23.44	67.74
30s	15s	6	83.57	43.24	18.03	65.07	80.42	52.25	31.15	67.03	86.71	31.53	59.02	69.65	83.92	53.15	18.03	67.69
30s	15s	7	80.58	40.91	21.67	62.95	83.81	52.73	31.67	69.20	85.25	42.73	56.67	70.98	82.73	50.91	20.00	66.52
30s	15s	8	79.85	49.54	10.71	63.47	79.12	55.05	30.36	66.89	86.81	39.45	53.57	70.78	83.15	51.38	25.00	67.81
30s	15s	9	79.32	50.00	12.50	63.32	80.83	51.89	35.71	67.76	71.80	64.15	0.00	60.51	86.47	48.11	21.43	68.46
30s	15s	10	79.62	51.89	11.54	64.11	80.00	56.60	32.69	68.18	69.62	69.81	5.77	61.72	83.85	50.94	25.00	68.18

Table 9: Full hyperparameter testing results.

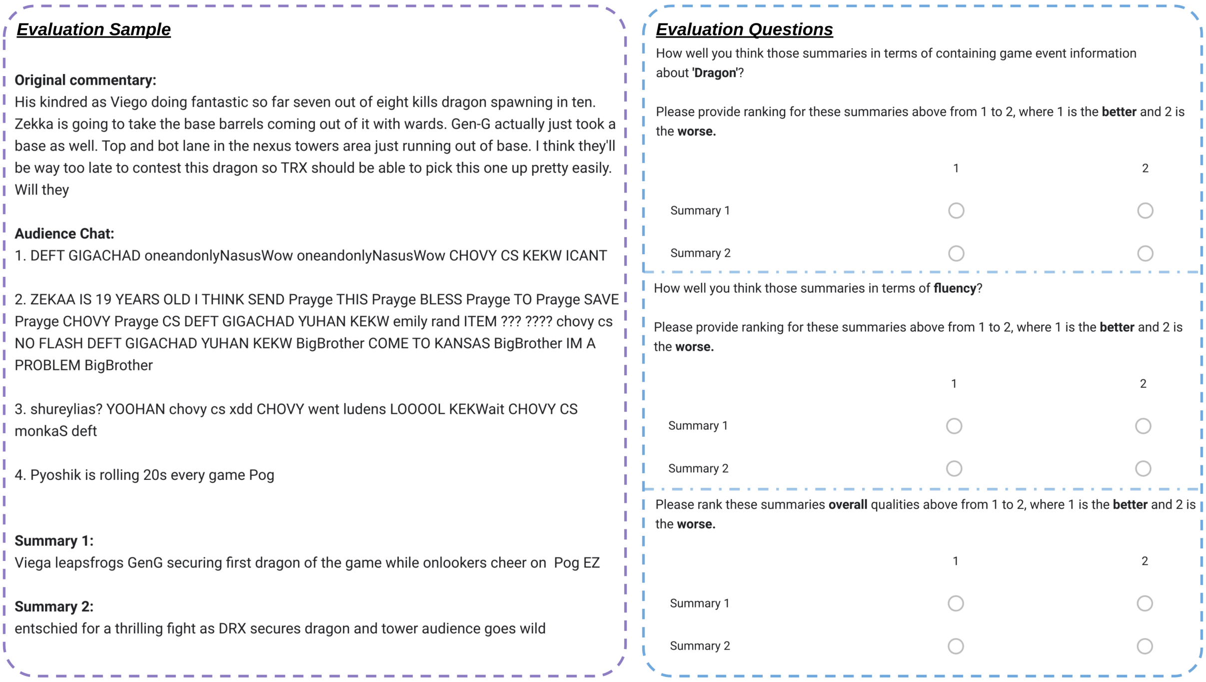

Figure 4: Screenshot of a human evaluation sample. Workers are shown the original commentary with truncated audience chats on the top left. We provide the generated summarisations on the bottom left. The worker ranks these two summarisations in terms of the inclusion of the game event, coherence and overall quality.