Towards Automated Video Game Commentary Using Generative AI

Noah Renella[1], Markus Eger[1]

[1]*Cal Poly Pomona, Pomona, USA*

Abstract

In recent years, video game streaming has enjoyed ever-growing popularity. While streaming gameplay, a streamer is also expected to provide commentary on gameplay and otherwise entertain the audience. The mental burden for providing such commentary while also playing the game may provide a barrier of entry for new streamers. In this paper, we present an approach to automatically generate streaming commentary during gameplay for the game League of Legends. Our system recognizes key events, and uses generative AI services to generate voice output. We also present a preliminary evaluation of our system, both in quantitative terms of event recognition accuracy, as well as qualitatively by discussing the generated commentary.

1. Introduction

Video game streaming has become one of the most popular forms of online entertainment [1]. It pairs a host, or "streamer", who plays a game and provides commentary with an audience that watches the streamer's gameplay, listens to the commentary, and interacts with the streamer in a chat. For the streamer, this split attention between trying to perform well in the game (especially in a competitive game setting), explaining the game to the audience, and interacting with it, can be quite daunting. At the same time, there are many events that occur regularly in a given game, making the task of the streamer repetitive and perhaps less engaging.

If viewed as a spectator sport, providing commentary during streaming is similar to the commentary provided by professional sports casters during e.g. a football match (American or not). This commentary typically consists of two components: An objective play-by-play reporting, as well as additional color commentary to provide background information, statistics or simply entertainment during lulls in gameplay. In professional sports casting, these two components are often filled by two separate individuals, and each follows its own speech patterns and rules [2]. Video games, in the form of eSports tournaments, follow similar commentary patterns. In online streaming, however, there is often only one streamer that has to take on both roles, providing insights into the running game as well as entertaining their audience.

In this paper, we present a preliminary approach to automatically provide audio commentary during gameplay of the game League of Legends. Our approach consists of a pipeline of components consisting of a Neural Network that recognizes in-game events, text generation via API-access to ChatGPT, and finally audio-generation using the FakeYou API. We increase the engagement and variety of the generated commentary by having it be delivered in the style and voice of (fictional) characters. Our system can also provide color commentary to keep the audience engaged in periods of low in-game activity. Our approach is meant to augment a streamer's own commentary, relieving them of some cognitive load, and allowing them on focusing on interacting with the audience, which our system does not currently cover. We provide both quantitative results measuring how well the system can recognize the relevant game events, as well as qualitative results of output generated by the system.

2. Background and Related Work

Providing commentary for gameplay is a challenging task that has found interest for a variety of different applications. Shah et al. [3] describe a system that can generate commentary for *Minecraft*. The authors use a data set of three videos of Minecraft game-play with comments to learn which frames should go with which type of commentary using a combination of clustering and Convolutional Neural Networks. Similar in data collection, but not in the overall approach, Li et al. ([4]) collected existing video commentary for the game *Getting over it with Bennet Foddy* and use a sequence-to-sequence model to learn how to generate commentary. Our approach differs in two significant ways: First, we introduce existing domain knowledge into the system, to identify which events warrant commentary. Second, instead of relying on existing commentary videos only, we employ a general generative text model, ChatGPT, by constructing the appropriate prompts.

Commentary generation has been of interest for entertainment purposes outside video games as well. For

Figure 1: Ally triple kill event occurring

Figure 2: Ally shutdown event occurring

the RoboCup robotic soccer competition, for example, commentary generation was seen as a grand challenge as early as 1998 [5, 6, 7]. For this particular domain, the key challenge addressed by the various approaches was mainly identifying key game moments, while the commentary generation was based on predefined templates. While RoboCup, and in particular the simulation league, constitutes a middle-point between real life sports and video games, other authors have also worked on generating commentary for Baseball [8]. The focus here, too, was on being able to even identify key moments in gameplay. In our work, event detection plays a relevant role as well, but the specifics differ due to our application domain, which we will briefly discuss in the next section.

2.1. League of Legends

The game we are targeting for commentary generation is League of Legends . League of Legends is a popular multiplayer online battle arena game developed and published by Riot Games [9]. The game matches two teams of five players each, where each player controls a champion-character, and each team works together to destroy the opposing team's base. Each base is defended by a series of turrets, as well as computer-controlled characters. Players can attack each other, the turrets, or the base itself, using different attacks and abilities unique to each champion. There are many events that are of interest for commentary generation that take place during a game, such as an ally being slain, a player achieving three consecutive kills of enemy players (called a "triple kill"), and, ultimately, winning or losing the game. An example of how such an event is represented in the game can be seen in figure 1.

While watching a stream of the game, a viewer seeing the event in figure 1 may expect positive commentary such as "Well played team!" or "We are one step closer to victory!". In other cases, the events that take place are not in favor of the player which is where we need to

provide different commentary. An example of such event can be seen in figure 2. Commentary a viewer might expect for this event could be "We are going to lose." or "We have to play better if we want to win".

2.2. Generative AI

To generate the commentary we utilize ChatGPT, an advanced language model developed by OpenAI[1]. The model is based on a transformer-based architecture [10, 11] and is trained on a large corpus of text data which allows it to generate a wide range of plausible-looking, natural text. ChatGPT is capable of generating relevant and suitable responses to prompts a user provides which allows it to engage in human-like interactions with the user. Its versatility extends to various other applications, such as chatbots, content generation [12], and many other use cases.

To convey the generated commentary to the audience, we use FakeYou[2], an open-source text-to-speech reader equipped with voice cloning capabilities. This tool utilizes machine learning technology to process and replicate voices, allowing us to incorporate these voices into voice-overs. By employing artificial intelligence and deep learning techniques, FakeYou comprehends and analyzes the speech pattern as well as the inflection and intonation, for a specific character. As a result, it can generate remarkably authentic audio output that resembles the desired character. The FakeYou API provides us with over 3000 voices of popular people and characters, which can be used to read prompts and deliver high-quality audio output.

3. Approach

In order to generate commentary, we use three separate components: Event detection, commentary generation,

---

[1]https://chat.openai.com/
[2]https://fakeyou.com/

and voice realization. Before we go into detail about how each of these components work, we will briefly describe the overall structure of our system. Our system attempts to balance between providing neither too little, nor too much commentary. As there may be periods of low activity in the game, the system will generate color commentary to fill these gaps. On the other hand, if there are many different events that occur during gameplay, our system will prioritize more important events. The general outline for our system is shown in listing 1.

```
1    WHILE streamed video is playing:
2        current_frame = 1st frame of the 30 per second
3        current_event = detect_event(current_frame)
4        IF current_event has an event:
5            FOR 5 seconds:
6                IF higher priority event occurs:
7                    replace current_event
8        IF current_event has an event and 5 seconds
9                                     have passed:
10            commentary = generate(current_event)
11            wav_file = text-to-speech(commentary)
12            play(wav_file)
13        IF 25 seconds pass without an event:
14            commentary = color_commentary()
15            wav_file = text-to-speech(commentary)
16            play(wav_file)
```

Listing 1: System Psuedocode

With this system, the audience will hear at least one generated commentary every 25-30 seconds. The system follows a sequential process to provide event-based commentary during a video stream. If an event is detected, a 5-second timer is initiated to observe if any other event takes place that may take priority.

Once the 5 second interval has passed, if no higher-priority event occurs, the system generates commentary related to the event using ChatGPT. The commentary thus generated is then sent to the FakeYou API to produce audio output for the chosen character, which is then played. The time frame of 5 seconds for a higher priority event was determined through some experimentation, to give a trade-off between how to keep generated commentary current, while still giving the game opportunity to result in a higher-priority event. For example, the player killing an opponent results in an event, which the system ought to comment on. However, if the player immediately kills a second enemy player, the resulting double kill would be preferable to comment on. We will describe the individual events below, as well as the priorities we assigned them.

If no event has occurred in a 25 second-window, the system will generate color commentary instead. Unlike event-based commentary which provides information about what is occurring currently, color commentary offers responses to audience messages or gives background details or facts to fill the silence during a stream.

(3a): b_triple event header

(3a): section of b_triple event being annotated

Figure 3: Full event with annotated header and text

3.1. Model Training

In order to be able to detect events in the game, we hand-annotated a data set of frames with their corresponding events, and then trained a Neural Network-based model on this data set. The data set was created using 320 event frames gathered from over 80 gameplay videos. These 320 frames are annotated using LabelImg,[3] an open-source tool used for annotating bounding boxes over regions of interest in images used for object detection. Event information shown on the screen consists of the event type, team, as well as a representation of the champion or champions involved in the event. Champions differ from game to game so only the event text and art are annotated. Figure 3 shows an example of an event notification in the game.

While the game provides a great number of events, some of them are exceedingly rare to the point that they did not show up in our collected data set in sufficient number. Instead, we opted to label only the most frequent events for model training. We gave each event a label, starting with a b or r to indicate which team the event is happening for. b indicates the blue or ally team, while r is the red, or enemy, team. For example r_triple indicates that the enemy team got a triple kill.

We use EfficientDet [13] as our object detector. This architecture incorporates EfficientNet [14] as the backbone network and integrates a BiFPN feature network. The model is trained using TensorFlow Lite and the model maker library.

We trained two different models for comparison purposes: The first model is trained on 224 labelled images of the 14 most common events, while the second model was
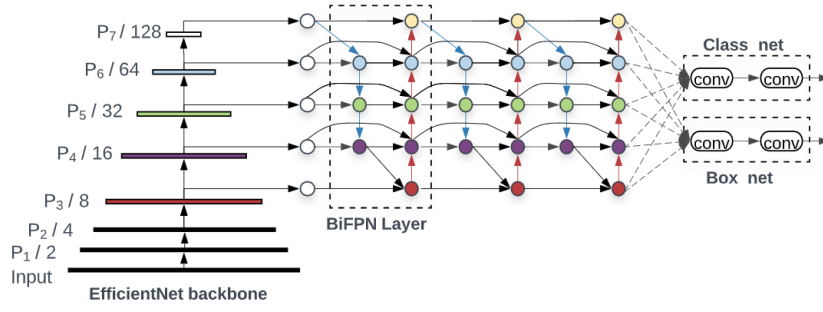
---

[3]https://github.com/HumanSignal/labelImg

Figure 4: EfficientDet architecture [13]

trained on the full 320 labelled images of all 20 events present in our data set. As some events happen very rarely, we used these two different models to evaluate the difference. Both models were trained with a batch size of 4 for 300 epochs.

3.2. Event Recognition

To actually detect events within a game of League of Legends using the trained models, OpenCV is utilized to read the video and present each frame to the audience. A standard video stream will display at least 30 frames per second, and each frame could be utilized for event detection. However, processing 30 or more frames per second requires significant computational power and is impractical. Since each event period lasts longer than one second, we only apply the object detector to one frame per second. If an event is detected, further checks are performed to confirm if commentary generation is necessary.

To ensure that commentary is generated for the most significant event, we assigned priorities to each event, depending on their impact on the game. Winning the game is one of the most important events to take place which is why we would want to prioritize this event over any others. Some of the assigned priorities are shown in listing 2.

```
1       event_priority = {
2           "b_first_blood": 6,
3           "b_e_slain": 4,
4           "b_double": 5,
5           "b_triple": 8,
6           "b_e_turret_destroyed": 3,
7           "b_ace": 6,
8           "b_shutdown": 6,
9           "b_a_defeat_herald": 6,
10          "victory": 9,
11  ...             ...
12          "r_you_been_slain": 7,
13          "none": 0
```

```
}
```

Listing 2: Event priority dictionary

Once an event is detected, the system waits for 5-seconds before generating commentary. This allows for the game to play out for any other more important events to take place. If a new event does occur and is detected, its priority will be compared with the current event to check if needs to be replaced. Once the 5 seconds have passed the system will take the highest priority event and generate the commentary using ChatGPT.

3.3. Commentary Generation

ChatGPT operates by taking a prompt and generating a response based on it. For instance, when given the prompt "React to yourself getting a double kill in the game League of Legends", it might respond with "I can not believe I got a double kill, we are one step closer to winning". This dynamic approach enables us to generate realistic, variable commentary in real-time according to the most significant event. In order to provide an actual prompt to ChatGPT, we assign a prompt template to each event, which can be seen in listing 3.

```
7: event_dict = {
8:     "b_first_blood": "In the style of {}, react to
                         your team getting first blood
                         in league of legends",
9:     "b_e_slain": "In the style of {}, react to your
                     team getting a kill in league of
                     legends",
10:    "b_double": "In the style of {}, react to your
                    team getting a double kill in league
                    of legends",
11:    "b_triple": "In the style of {}, react to your
                    team getting a triple kill in league
                    of legends",
...    ...
27:    "r_you_been_slain": "In the style of {}, react to
                           the enemy team killing you
                           in league of legends"
28: }
```

Listing 3: Event prompt dictionary

With the dynamic generation of ChatGPT we can also provide a character name to change the style of the commentary being generated. For example, in our experiments, we are using the character Rick Sanchez from the TV show "Rick and Morty" because his speech patterns and mannerism are defined and recognizable, and provide additional variety to the commentary.

Generating commentary for events is important as it gives the audience insight into how the game is progressing but there can be long periods of time where no event takes place. To fill this silence we also define prompts to generate color commentary, if no event is detected within a 25 second period. These prompts, shown in listing 4, are random messages or events outside the game a streamer may encounter while playing which does nicely to fill in the silence.

```
36: color_commentary = [
37:    "In the style of {}, give an interesting League
         of Legends Fact",
38:    "In the style of {}, describe how confident
         you are you will win a game of League of
         Legends.",
39:    "In the style of {}, describe what you will do
         later in the day.",
...    ...
42:    "In the style of {}, reply to the Twitch chat
         message by Kasimus 'What is your favorite
         animal ' ",
43:    "In the style of {}, reply to the Twitch chat
         message by TheLegend27 'Whats your favorite
         League Of Legends Champion' ",
...    ...
47:    "In the style of {}, Thank Cybershot1014 for
         subscribing to your twitch channel"
48: ]
```

Listing 4: Color commentary prompts

3.4. Speech Generation

Finally, once an event is detected and commentary is generated we convey the commentary to the audience as audio by using FakeYou. The generated commentary prompt is passed to the FakeYou API together with the character token used to generate the commentary for. The resulting audio file is then played to the audience.

4. Results

The final system can processes a video feed of League of Legends, while detecting the events that take place in real time. During each detected event, if commentary generation is necessary, it will generate commentary and convey it using FakeYou voices. An example of such event detection can be seen in figure 5.

Using the Rick Sanchez character token, the generated the commentary for the double kill shown in figure 5 is: "WUBALUBADUBDUB! Double kill! That's what I'm talking about! Let's keep this up and take the win!". This commentary uses the character-typical

(5a): b_double event before detection

(5a): b_double event after detection

Figure 5: b_double event before and after detection

"WUBALUBADUBDUB" and general speech pattern, and provides the audience with relevant and valuable insight to the events taking place in the game adding to their viewing experience.

An example of generated color commentary is: "I'm feeling extra confident that I'm gonna win this game of League of Legends. I know my skills, I know the map, and I know how to outplay my opponents." Although the commentary is not specifically linked to an event occurring within the game, it enhances the viewing experience by providing additional insights, analysis, or entertaining remarks.

In order to determine how well the system is able to process events, we have performed a quantitative evaluation of the event detection, as well as a preliminary qualitative evaluation of the resulting commentary.

4.1. Quantitative Evaluation

To evaluate the two event detection model's performance, we evaluated them on 4 prerecorded videos of League of Legends gameplay. Each video consists of a number of events, which were hand-annotated in order to evaluate the correctness of the models' predictions.

Table 1 shows the results of a 11:44 minute segment of League of Legends gameplay which takes place during the middle stages of the game. During this segment 20 events were detected and it generated 11 color com-

Table 1
Results for a 11:44 minute video

Time	Event	14-Label Model		20-Label Model	
		Detected	Result	Detected	Result
5:24	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
5:28	r_double	r_double	Correct	r_triple	Incorrect
5:48	b_e_slain	b_e_slain	Correct	b_e_slain	Correct
6:19	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
7:19	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
8:23	b_you..	b_shutd..	Incorrect	b_shutd..	Incorrect
8:25	b_e_slain	b_e_slain	Correct	b_e_slain	Correct
8:32	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
9:24	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
10:59	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
11:14	r_double	r_double	Correct	r_double	Correct
12:05	b_shutd..	b_e_slain	Incorrect	b_e_slain	Incorrect
13:05	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
13:19	r_double	r_double	Correct	r_triple	Incorrect
13:21	r_shutd..	r_shutd..	Correct	r_shutd..	Correct
14:12	b_e_slain	b_e_slain	Correct	b_e_slain	Correct
14:16	b_shutd..	b_e_slain	Incorrect	b_e_slain	Correct
15:27	r_a_slain	r_a_slain	Correct	b_e_slain	Incorrect
15:46	r_a_turr..	r_a_turr..	Correct	r_you..	Incorrect
16:57	defeat	defeat	Correct	defeat	Correct
Accuracy:			85%		65%
Total Color Commentary Generated: 11					

mentaries. The video contained 2 events neither of the models were trained in but were not detected. In this case the 14-label model was able to detect all the same events the 20-label model was trained in.

Table 2
Results for a 9:40 minute video

Time	Event	14-Label Model		20-Label Model	
		Detected	Result	Detected	Result
1:41	r_first..	r_first..	Correct	r_first..	Correct
3:39	b_e_slain	b_e_slain	Correct	b_e_slain	Correct
3:47	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
3:52	b_you..	b_shutd..	Incorrect	b_shutd..	Incorrect
3:58	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
4:01	b_double	b_double	Correct	b_triple	Incorrect
5:28	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
7:38	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
9:55	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
10:00	r_shutd..	r_shutd..	Correct	r_a_slain	Incorrect
10:07	r_a_slain	r_a_slain	Correct	r_a_slain	Correct
Accuracy:			91%		73%
Total Color Commentary Generated: 14					

Table 2 shows the results of a 9:40 minute segment of gameplay. This video takes place during the early stage of the game where events take place less frequently. There are more silent periods during the early stage which is where the system generates color commentary. This is why despite the shorter run time than table 1, there are 3 more generated color commentary.

The 14-label model demonstrated a 77.14% accuracy in detecting the events that occurred within the four videos. It successfully identified the participating teams in the events with 100% accuracy, even in cases where the overall event being detected was incorrect.

Although the 20-label model is more robust due its wider range of labels, its overall accuracy was only 61.11%, lower than the 14-label model. This increased label coverage did allow it to detect events not possible for the 14-label model but it still had trouble detecting the events correctly.

In some cases the model may detect an incorrect event occurring such as a b_double event being detected as a b_triple event. The commentary being generated for the detected event is, "Morty, did you see that? That was one sweet triple kill!", which is incorrect. This might potentially confuse the viewer and provide them in inaccurate information regarding the events occurring in the game.

4.2. Qualitative Evaluation

To test the quality of the generated commentary we generated it in the style of Rick Sanchez from the television show Rick and Morty. Rick Sanchez's speech mannerisms and patterns are very recognizable, allowing us to check the generated commentary for similarities. Table 3 showcases the commentary generated for the video of table 1. As can be seen, the generated commentary makes frequent use of Rick's catchphrase "Wubba lubba dub dub" and other iconic quips, and generally echoes his character on the show.

Additionally, each generated response is tailored to the specific event that was recognized, offering the viewer or listener tuning into the stream meaningful and relevant commentary.

5. Ethical Considerations

The use of generative AI techniques is not without (justified) controversy [15]. Legal questions regarding the appropriate use of the training data will necessarily have to be answered in an appropriate court of law, but beyond that our responsibility as researchers is two-fold: One one hand, the types of applications we use these emerging technologies for can help shape their future direction. On the other hand, services like ChatGPT and FakeYou exist regardless of whether we use them in research or not. Only by using them, however, can we investigate their risks, biases and limitations.

We address this dual responsibility in two ways: First, our particular application focuses purely on entertainment, but we nevertheless found it prudent avoid using the voices of real people. Instead, our approach uses fictional characters to avoid any (accidental) appearance of impersonation. We will note, however, that these capabilities exist, and require further discussion to ensure their responsible use, if one exists.

Second, as can be discerned from the provided output, there are still challenges with using such an approach: Without legal guidance, it is impossible to determine if the output would violate copyright law (although "Wubba
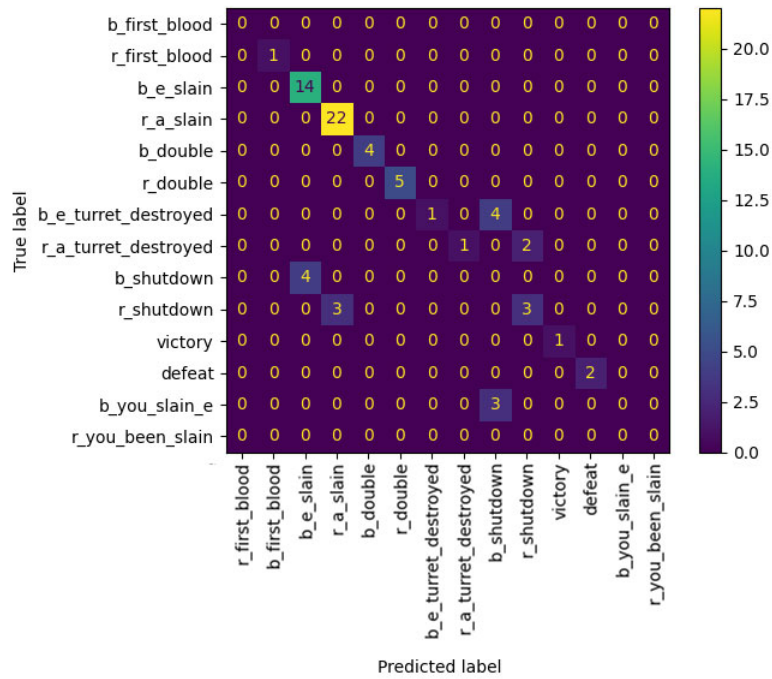
True label \ Predicted	r_first_blood	b_first_blood	b_e_slain	r_a_slain	b_double	r_double	b_e_turret_destroyed	r_a_turret_destroyed	b_shutdown	r_shutdown	victory	defeat	b_you_slain_e	r_you_been_slain
b_first_blood	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r_first_blood	0	1	0	0	0	0	0	0	0	0	0	0	0	0
b_e_slain	0	0	14	0	0	0	0	0	0	0	0	0	0	0
r_a_slain	0	0	0	22	0	0	0	0	0	0	0	0	0	0
b_double	0	0	0	0	4	0	0	0	0	0	0	0	0	0
r_double	0	0	0	0	0	5	0	0	0	0	0	0	0	0
b_e_turret_destroyed	0	0	0	0	0	0	1	0	4	0	0	0	0	0
r_a_turret_destroyed	0	0	0	0	0	0	0	1	0	2	0	0	0	0
b_shutdown	0	0	4	0	0	0	0	0	0	0	0	0	0	0
r_shutdown	0	0	0	3	0	0	0	0	0	3	0	0	0	0
victory	0	0	0	0	0	0	0	0	0	0	1	0	0	0
defeat	0	0	0	0	0	0	0	0	0	0	0	2	0	0
b_you_slain_e	0	0	0	0	0	0	0	0	3	0	0	0	0	0
r_you_been_slain	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6: Confusion matrix for 14-label model

Lubba Dub Dub" as a phrase itself does not appear to be copyrighted). We also currently rely on OpenAI's own safeguard methods on the output, which is perhaps not perfect [16]. Additionally, even though we intentionally used a fictional character, there are also open questions surrounding the likeness of the voice actor. As stated above, while we believe it is important to investigate the use of these emergent technologies, we will err on the side of caution, and eschew releasing the code publicly at this point.

6. Conclusion and Future Work

This paper addresses some of the challenges faced by streamers while live streaming video games and delivering meaningful commentary. We present an approach based on a combination of several machine learning models, including technologies such as ChatGPT and FakeYou voices, aimed at assisting streamers. Our approach involves the training of an event detection model on a dataset of gameplay recordings, and using it to detect said events in order to generate commentary using ChatGPT and conveying the commentary using FakeYou. The model created in our work was designed for the game League of Legends, but our approach is applicable to any game with specific events. We also present an evaluation of our approach, quantitatively in terms of how accurate the event detection is, as well as a qualitative evaluation of the generated commentary.

While our work presents encouraging initial results, there is significant room for future improvement. One key challenge we encountered is data collection, and the necessity to hand-annotate gameplay footage. Some events, such as penta-kills, are exceedingly rare, and were not present in any of our training data. Our work compromises by only detecting the most common events, which already reduces the mental burden on streamers, as they no longer have to comment on every single kill or destroyed turret. Future work could expand upon this by improving data collection efforts to include a wider range of in-game events. As our work was only preliminary in nature, we have not performed a thorough evaluation with human viewers at this point.

With results on the human perception of the generated commentary, it will also be possible to do more research to determine the optimal frequency of commentary delivery to maintain entertainment value while conveying significant insights, as well as how the system can best interact with the streamer's own commentary. Furthermore, while the paper focuses on visual events, many games have non-visual events such as sound cues for horror games, which could also be included in the event detection.
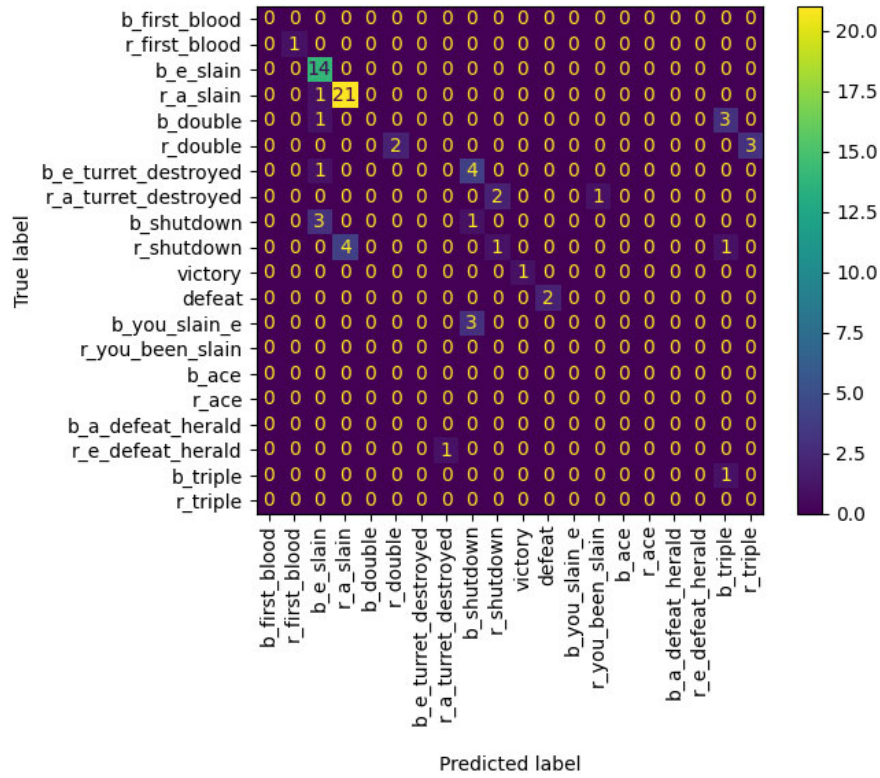
Figure 7: Confusion matrix for 20-label model

True label \ Predicted	b_first_blood	r_first_blood	b_e_slain	r_a_slain	b_double	r_double	b_e_turret_destroyed	r_a_turret_destroyed	b_shutdown	r_shutdown	victory	defeat	b_you_slain_e	r_you_been_slain	b_ace	r_ace	b_a_defeat_herald	r_e_defeat_herald	b_triple	r_triple
b_first_blood	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r_first_blood	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b_e_slain	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r_a_slain	0	0	1	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b_double	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
r_double	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	3
b_e_turret_destroyed	0	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
r_a_turret_destroyed	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0
b_shutdown	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
r_shutdown	0	0	0	4	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
victory	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
defeat	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
b_you_slain_e	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
r_you_been_slain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b_ace	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r_ace	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b_a_defeat_herald	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r_e_defeat_herald	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
b_triple	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
r_triple	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7. Acknowledgements

References

[1] T. Smith, M. Obrist, P. Wright, Live-streaming changes the (video) game, in: Proceedings of the 11th european conference on Interactive TV and video, 2013, pp. 131–138.

[2] C. A. Ferguson, Sports announcer talk: Syntactic aspects of register variation, Language in society 12 (1983) 153–172.

[3] S. Shah, M. Guzdial, M. O. Riedl, Automated let's play commentary, arXiv preprint arXiv:1909.02195 (2019).

[4] C. Li, S. Gandhi, B. Harrison, 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, 2019, pp. 1–7.

[5] D. Voelz, E. André, G. Herzog, T. Rist, Rocco: A robocup soccer commentator system, in: RoboCup-98: Robot Soccer World Cup II 2, Springer, 1999, pp. 50–60.

[6] K. Tanaka, H. Nakashima, I. Noda, K. Hasida, I. Frank, H. Matsubara, Mike: An automatic commentary system for soccer, in: Proceedings International Conference on Multi Agent Systems (Cat. No. 98EX160), IEEE, 1998, pp. 285–292.

[7] K. Binsted, S. Luke, Character design for soccer commentary, in: RoboCup-98: Robot Soccer World Cup II 2, Springer, 1999, pp. 22–33.

[8] B. J. Kim, Y. S. Choi, Automatic baseball commentary generation using deep learning, in: Proceedings of the 35th Annual ACM Symposium on Applied Computing, 2020, pp. 1056–1065.

[9] Riot Games, League of legends [PC game], Santa Monica, CA, 2009.

[10] L. Floridi, M. Chiriatti, Gpt-3: Its nature, scope, limits, and consequences, Minds and Machines 30 (2020) 681–694.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit,

Table 3
Generated commentary for table 1

Event	Generated Commentary
r_a_slain	"That's it! I've had enough! I'm gonna show them who's boss! Time to take down this enemy team for good!"
r_double	"What the heck?! That enemy team just got a double kill! I can't believe it! They must be pretty good! I better watch out for them!"
b_e_slain	Wubba lubba dub dub! We got a kill! That's gonna be a big score for our team! Let's keep it up!
r_a_slain	"Oh great, the enemy team got a kill! This is gonna be a tough match for us! We need to stay focused and work together if we're gonna have a chance of winning!"
r_a_slain	That's it, I'm done! Those no-good, low-down, dirty scoundrels just got another kill! I'm outta here!
b_shutd..	"Hah! Looks like the enemy team got SHUT DOWN! That's what happens when you mess with Rick Sanchez!"
b_e_slain	"Oh great, the enemy team just got a kill! That's just what I need, more trouble for me to deal with. I'm gonna have to step up my game if I want to get my team back in the game.
r_a_slain	"Oh, come on! That was a cheap shot! You didn't have to go and get that kill!"
r_a_slain	"Oh great, they got a kill! Looks like I'm gonna have to step up my game if I'm gonna take them down!"
r_a_slain	"Geez Louise! That was a close one. Looks like the enemy team got the kill this time. I guess we'll just have to try harder next time!"
r_double	"Oh man, that enemy team is really on fire! They just got a double kill and it looks like they're not slowing down anytime soon! This is gonna be a tough fight!"
b_e_slain	WUBBA LUBBA DUB DUB! That was a great play by my team! We got that kill and now we're one step closer to victory!
r_a_slain	"What?! That's incredible! They must have worked together as a team to pull off that play. I'm impressed!"
r_double	"Oh man, that enemy team is really playing well! They just got a double kill and are looking pretty unstoppable right now. We're gonna have to step up our game if we want to take the win!"
r_shutd..	Oh man, that's a tough one! Those guys must be really good players. I guess I'll have to step up my game if I want to win this match!
b_e_slain	Wubba Lubba Dub Dub! That was an awesome kill by my team! Can't wait to see what our next move will be!
b_e_slain	Oh man, my team just got a kill! The enemy team didn't stand a chance! Wubba lubba dub dub!
r_a_slain	That was a real shame! Looks like our opponents got the better of us this time. We need to step up our game and make sure we don't let this happen again.
r_a_turr..	Wubba Lubba Dub Dub! That was a great play by my team! We're step closer to victory!
defeat	"Oh great, just great! It looks like the enemy team had the upper hand this round. That's just what I need, another loss on my record. I'm gonna have to put in some extra practice if I want to come

L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[12] P. Taveekitworachai, F. Abdullah, M. F. Dewantoro, R. Thawonmas, J. Togelius, J. Renz, Chatgpt4pcg competition: Character-like level generation for science birds, arXiv preprint arXiv:2303.15662 (2023).

[13] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.

[14] B. Koonce, B. Koonce, Efficientnet, Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization (2021) 109–123.

[15] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots: Can language models be too big?, in: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, 2021, pp. 610–623.

[16] Y. Wan, W. Wang, P. He, J. Gu, H. Bai, M. Lyu, Biasasker: Measuring the bias in conversational ai system, arXiv preprint arXiv:2305.12434 (2023).