Generating Racing Game Commentary from Vision, Language, and Structured Data

Tatsuya Ishigaki[†] **Goran Topić**[†] **Yumi Hamazono**[†◦] **Hiroshi Noji**[†◇]
Ichiro Kobayashi[†◦] **Yusuke Miyao**[†‡] **Hiroya Takamura**[†]

[†]National Institute of Advanced Industrial Science and Technology, Japan,
[◦]Ochanomizu University, [‡]The University of Tokyo, [◇]LeepMind Inc.
{ishigaki.tatsuya, goran.topic, hamazono-yumi, noji, takamura.hiroya}@aist.go.jp,
koba@is.ocha.ac.jp, yusuke@is.s.u-tokyo.ac.jp

Abstract

We propose the task of automatically generating commentaries for races in a motor racing game, from vision, structured numerical, and textual data. Commentaries provide information to support spectators in understanding events in races. Commentary generation models need to interpret the race situation and generate the correct content at the right moment. We divide the task into two subtasks: utterance timing identification and utterance generation. Because existing datasets do not have such alignments of data in multiple modalities, this setting has not been explored in depth. In this study, we introduce a new large-scale dataset that contains aligned video data, structured numerical data, and transcribed commentaries that consist of 129,226 utterances in 1,389 races in a game. Our analysis reveals that the characteristics of commentaries change depending on time and viewpoints. Our experiments on the subtasks show that it is still challenging for a state-of-the-art vision encoder to capture useful information from videos to generate accurate commentaries. We make the dataset and baseline implementation publicly available for further research.[1]

1 Introduction

Live commentary plays an important role in sports matches and video games; it makes spectators more excited, more immersed, and better informed about the matches or games (e.g., Schaffrath (2003)), as in the example of racing game commentary "*We are approaching the final long straight. I wonder who is going to win!*". Live commentary also enhances the value of online videos and home videos. However, providing a live commentary requires a certain level of commenting skills and knowledge of the target sports or video games; the majority of online videos and home videos are left without live commentary.[2] The application of natural language generation technology would be a solution to this problem. Thus, we select the racing game domain as an example, and propose a task of generating live commentary on it.

Examples of utterances in a live commentary for a racing game are shown in Figure 1. Live commentaries should describe each important event in the race at the moment when the event occurs, within a short period of time. Thus, we have to make decisions on *when to speak* and *how long/elaborately to speak*, in addition to *what to say* and *how to say it*, which have been long studied. The importance of each event would have to be assessed in the context of a competition in which participants are striving to win. It suggests that *what to say* for race commentary generation should be different from, for example, image captioning. In this sense, the task of live commentary generation contains inherent limitations that are not addressed in well-studied generation problems such as summary generation for basketball (Puduppully and Lapata, 2021), and image/video captioning (Vinyals et al., 2015; Yao et al., 2015); however, some techniques developed for such existing generation tasks are also useful for live commentary generation.

As input, vision data, such as the videos shown in Figure 1, are common in many tasks. However, it is not a trivial task to capture important information from vision data, because many of the frames in a race would be similar to each other, unlike images for image captioning data. Therefore, we propose the use of structured data, that is, telemetry data, including the positions and the speeds of cars, and the steering wheel angles. The

[1] https://kirt.airc.aist.go.jp/RacingCommentary

[2] For example, many gameplay videos on Twitch do not have live commentary (https://www.twitch.tv).

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 103–113,
Aberdeen, Scotland, UK, 20-24 September 2021.

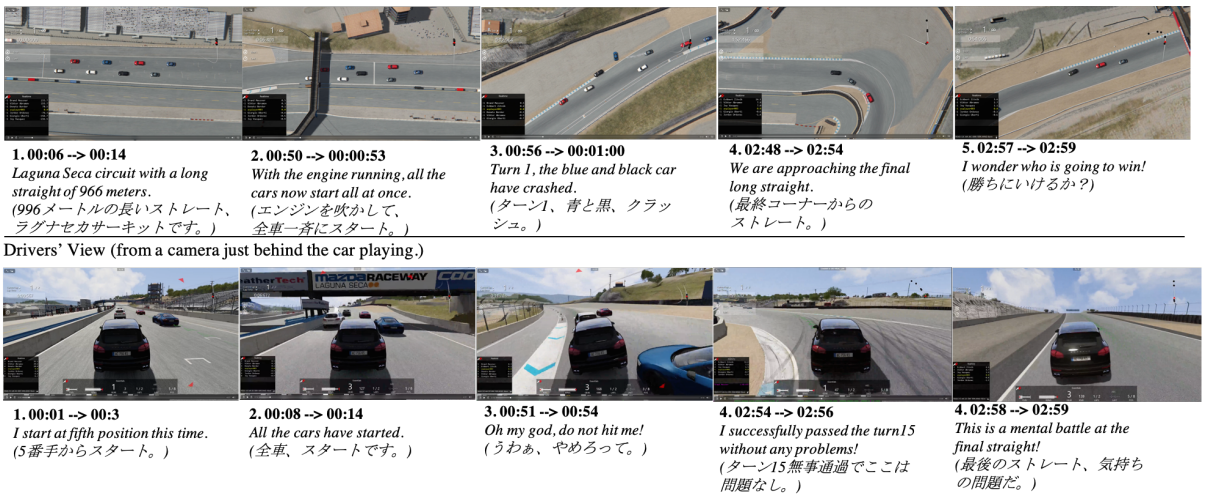

Figure 1: Translated Examples of commentaries (original utterances in Japanese are in brackets). For the commentaries on the top, the commentator is watching the race from the aerial viewpoint. For those at the bottom, the commentator is watching the race through a camera just behind the driver.

assumption that such telemetry data are available is not unrealistic. It is the general trend that many sensors are used to obtain telemetry data even in real sports matches and motor races. For example, each race car in F1 races is monitored by 300 sensors.[3] Additionally, players are tracked by GPS technology during football matches to obtain positional data (Memmert and Raabe, 2018). We work on video games because telemetry or vision data are easier to obtain than in real sports. This can address the huge demand in the gaming community and serve as a favorable test bed for live commentary generation. As a result, the task addressed in this paper is the live commentary generation for racing games from vision, structured, and textual data.[4]

Live commentary generation has not been studied in-depth, partly because of the lack of datasets. Thus, we create a new dataset for live commentary generation, which includes 129,226 utterances of live commentary, aligned with gameplay video and telemetry data of racing game. The telemetry data contain the positions and speeds of race cars and various types of information about circuits and cars. There are two types of live commentary. One is provided by the game players while playing and watching the racing game from the virtual camera behind the car. The other is provided by another person watching the game from a virtual helicopter. We analyze the differences in the characteristics of commentaries from different viewpoints.

We split the live commentary generation into two subtasks: the utterance timing identification and utterance generation. We propose multimodal models for these subtasks and also provide an empirical evaluation. Our experiments suggest that the use of telemetry data works well for this task, whereas it is difficult for a state-of-the-art vision encoder to extract useful information from race videos, especially for utterance generation.

Our contributions are threefold: (1) we propose a novel task of automatically generating motor racing game commentaries, (2) we create a dataset and analyze its characteristics, and (3) we propose methods for this task and argue that combining multimodal data is challenging, which is worth exploring in depth. We make the dataset and baseline implementations publicly available to enhance further studies on this task.

2 Related Work

Existing studies on commentary generation can be divided into real-time commentary (Taniguchi et al., 2019; Kim and Choi, 2020) and commentaries written afterwards (Puduppully and Lapata, 2021). Our focus is on the former. Live commentary generation is formulated as the extraction of tweets (Kubo et al., 2013), the combination of rules and keyword extraction from videos (Kim and Choi, 2020) and neural network-based data-to-text (Taniguchi et al., 2019). To

[3] https://aws.amazon.com/f1/

[4] We include textual data as input because we use the previous utterances as additional input.

generate commentary in real-time, we need to solve at least two tasks: timing identification and utterance generation tasks. However, existing studies focus on the latter, where the timings are given, for example, minute-by-minute updates (Kubo et al., 2013). Unlike baseball, the timing identification task for race commentary is not trivial because a race cannot be segmented simply.

Our setting can be considered as a combination of two different research topics: video captioning (Kim and Choi, 2020) and data-to-text (Taniguchi et al., 2019). Various methods for encoding video frames have been actively studied (Dosovitskiy et al., 2021); commentaries often include comments that focus on the positional relation between cars, which requires a more fine-grained understanding of video frames. The performance of current vision encoders still needs to be evaluated. Data-to-text is the task of converting structured data into natural language, which has been applied to the domain of finance (Murakami et al., 2017; Aoki et al., 2018, 2021; Uehara et al., 2020), weather forecast (Murakami et al., 2021), a summary of sports matches (Puduppully and Lapata, 2021; Iso et al., 2019) and live sports commentary (Taniguchi et al., 2019). The inputs used for existing studies are time-sequence numerical data (Murakami et al., 2017), tables (Puduppully and Lapata, 2021; Gardent et al., 2017) or simulated images (Murakami et al., 2021). These models focus on neural network-based approaches; however, data-to-text tasks have been studied for a long time (see a survey paper (Gatt and Krahmer, 2018) for details).

Existing studies on generation mostly focus on generating text from a single viewpoint, i.e. they generate objective descriptions of video frames in video captioning, and a data-to-text model generates a text that focuses on the main content of the input data. A few existing studies state that live commentaries change depending on the viewpoints of commentators. For example, Kubo et al. (2013) found that the generated commentaries on soccer matches are not objective, and these are biased to mention more popular teams. The viewpoints are the key to characteristic commentaries, but most studies have ignored the difference caused by the viewpoints that our dataset addresses.

Datasets play important roles in studies on generation. Existing datasets for generation tasks contain data in a single modality, such as, videos (Zhou et al., 2018; Krishna et al.) or structured data (Puduppully and Lapata, 2021; Gardent et al., 2017). We propose a new large-scale dataset that contains transcribed commentaries aligned with videos and structured numerical data.

3 Dataset

We describe the procedure used to create our dataset. We then show its statistics and the analysis to characterize the task.

3.1 Procedure for creating our dataset

Collecting recordings and spoken commentaries: We hired five workers who regularly play e-sports games. Thus, some of the workers are familiar with playing racing games, but some are not. They are not professional commentators. As a racing game, we used Assetto Corsa[5]. For each race consisting of two laps, one worker plays while simultaneously commentating it from the viewpoint of the virtual camera just behind the car (*driver's view*). Another worker is assigned to commentate the race from the viewpoint of a virtual helicopter (*aerial view*), without playing the game. Note that the commentaries are in Japanese. Drivers used a physical steering controller to achieve a situation close to real sports competitions.

For both commentaries, we ask the commentators to mainly mention the car driven by the player; however, the commentators could also mention other cars if they found them worth mentioning. Circuit maps, in which each turn is numbered, are available to commentators so that they can refer to them by numbers (e.g., *Turn 15*). Well-known turns or straights are given names such as *Casanova* for turn six in the Laguna Seca circuit.[6]

Collecting transcriptions of commentaries: After the collection of recordings, we hired 149 workers on a crowdsourcing service, Lancers[7], to transcribe all the recordings. Workers are supposed to transcribe the recordings and add the

[5] Assetto Corsa is a game title developed and published by Kunos Simulazioni:
http://www.kunos-simulazioni.com

[6] The numbers and names are obtained from Wikipedia or other websites that describe circuits.

[7] http://lancers.jp

Telemetry data types	Example values
current lap [0..]	1
is current lap invalidated?	false
lap time of current lap (ms)	256
lap time of previous lap (ms)	156164
progress on current lap [0, 1]	0.002780
projected diff. from best lap	0.0
speed (km/h)	177.693130
steer rotation (rad)	-59.793526
world position (x, y, z) (m)	(5.372770, 64.056038, -749.219971)
position on track (L=−1, R=1)	-0.515301
distance from ideal path (m)	0.854022

Table 1: List of collected structured telemetry data with example values. The last two types of data are not from the API, but are calculated by the authors.

# of unique circuits	4
# of commentators	5
total # of races	1,389
total # of recordings	2,473
- driver's viewpoint	1,389
- aerial viewpoint	1,084
total recording duration	226:37:53
- driver's viewpoint	126:11:19
- aerial viewpoint	100:26:34
total # of utterances	129,226
avg. # of utterances per race	52.25
avg. # of characters per utterance	22.22
avg. length of an utterance	2.73s
avg. length of silence	3.46s

Table 2: Statistics of the dataset.

start and end timestamps to each utterance. Utterances are basically sentences, with some exceptions; some utterances do not form complete sentences because they are truncated owing to speech repair. Finally, we manually checked whether the transcriptions aligned with the videos correctly.

Collecting structured telemetry data: We also collected structured telemetry data. Using Assetto Corsa's API, we extracted various structured numerical data, including the speeds of the cars participating in the race, % of the progress over the entire race, the angles of the steering wheel, and other 13 types of numerical values. The full list of the types of structured data collected is shown in Table 1. We repeated the extraction of these values every 0.01 seconds on average.

3.2 Statistics and Analysis

In total, the five workers had played 1,389 races. 1,084 out of the 1,389 races are given commentaries from both the drivers' and aerial viewpoints. The remaining 305 races are given only commentaries from the drivers' viewpoints. Thus, we collected a total of 2,473 video recordings aligned with commentaries and multimodal data.[8] The total duration of the recordings is 226 hours, and the number of collected utterances is 129,226, which is more than the number of descriptions in ActivityNet Captions dataset (Krishna et al.), a large dataset for dense video captioning. Also, as a non-English dataset, it might provide some valuable linguistic diversity, as most datasets are in English. On average, they produced an utterance with a length of 2.73 seconds and then they kept silent for 3.46 seconds. The other statistics are listed in Table 2.

We manually designed labels for the utterances to capture their characteristics. Each label is defined as a pair of two sub-labels, target label and the content label, as presented in Table 3.

The target label represents the target subject of the utterance, such as the `player's car`, `other cars`, `all cars`, or `the circuit`. For example, the utterance *"All the cars now start"* is labeled as `all cars`, because it focuses on all the cars participating in the race, whereas *"The player is now approaching Turn15"* is labeled as the `player's car`, because it mentions only the target car. The content label represents the content of the utterance, such as the `relative position`, `movement`, `lap time` and other content types as presented in Table 3. As an example, *the player is now approaching Turn15* is labeled as the *player's car:movement*, because it mentions the movement of the player's car.

We randomly extracted 874 utterances from 20 videos, and then manually annotated them using the listed labels. It should be noted that this manual labelling task is performed under the multi-label setting, which allows us to assign one or more labels to an utterance. We analyze the distributions of the labels to capture the characteristics of the dataset. In this analysis, we are particularly interested in (1) how the label distribution changes over time, and (2) how the label distribution differs depending on the commentator's viewpoint.

How utterances change over time?

We split a race into quarters according to the timeline (e.g., the first quarter corresponds to the interval from the beginning of a race to the 25% point).

[8] 1,084+305 videos from driver's viewpoints, and 1,084 videos from aerial viewpoints.

Target labels	Example utterances
player's car	*This was a very elegant overtake by the player.*
other cars	*The car behind just overtook the player.*
all cars	*All the cars has now started.*
circuit	*Laguna Seca is well known for its very long strait.*
none	*Ah!*
Content labels	**Example utterances**
relative position	*The player is now at the second making the distance close to the first.*
location on map	*The blue car is approaching Turn2 and others follow.*
lap time	*The player now crossed the finish line at the time 3.15*
previous event	*Maybe this mistake might cause big impact on the time*
future event	*Can the player successfully pass the difficult Turn 15?*
movement	*The player overtook the red* car on this long straight.
stable race	*All cars are stable without any problems.*
features	*All the cars are the same, Porsche Macan.*
greetings	*Ok, now I start my commentary on this race.*
reaction	*Oh!*
others	—

Table 3: List of sub-labels and example utterances. A label assigned to an utterance is defined as a pair of Target and Content sub-labels.

Figure 2 shows the label distributions for different quarters. In the figure, the proportions of the labels in the first quarter are represented by the tops of the four bars, which are colored blue. Similarly, the second (orange), third (black), and fourth (yellow) bars from the top represent the proportions in the second, third, and fourth quarters, respectively.

For the first quarter indicated by the top bar for each label, which are colored blue in the figure, the labels with *features* (i.e., *circuit:features*, *player's car:features* and *other cars:features*) are frequent compared with the other quarters. This suggests that commentators often start the commentary by mentioning the features of the circuit or the cars.

For the final quarter indicated by the bottom bars, which are colored yellow, *none:greetings* and *player's car:lap time* are frequent, suggesting that the commentators mention the elapsed time after the cars crossed the finish line and finally concluded the session with greetings.

Next, we focus on the differences between the two middle quarters, indicated by the second and third bars, which are colored orange and black.

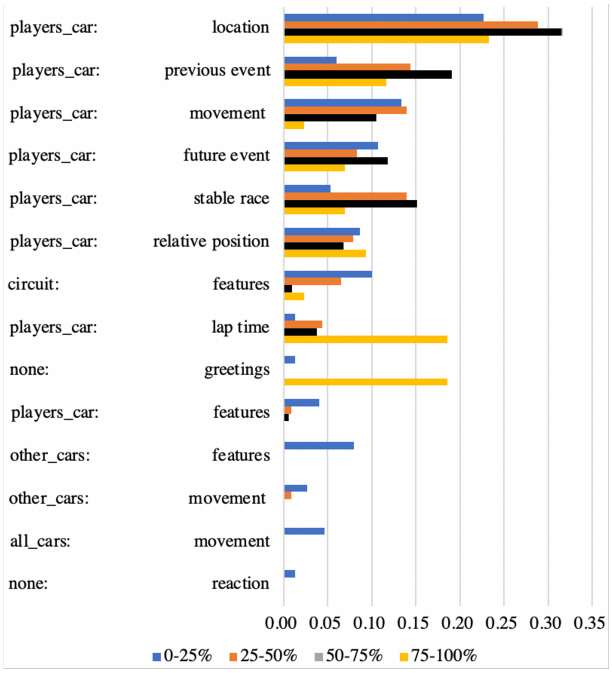

Figure 2: Distribution over utterance labels in different periods of timestamps

The proportion of *player's car:movement* in the second quarter (the second bars colored orange), is higher than in the third quarter (the third bars colored black). Thus, the commentators tend to mention more facts in the second quarter. In contrast, the third quarter (the third bars colored black) contains more *future event* and *previous event* labels that often include commentators' comments, concerns, or opinions on the previous and future events. This suggests that there are more mentions on the objective facts in the early stages of races, whereas subjective utterances increase toward the end of the races.

How do utterances differ depending on the viewpoint?

We examined the differences between commentaries from two different viewpoints: the driver's and aerial. Figure 3 shows the label distribution, where the upper bars colored orange correspond to the driver's viewpoint, and the lower bars colored blue correspond to the aerial viewpoint.

The proportion of the *player'car:location* for the aerial viewpoint is almost double of that of the proportion of the same label for driver's viewpoint. This is because the commentators with aerial viewpoint can capture the locations in maps more easily, whereas commentators with driver's

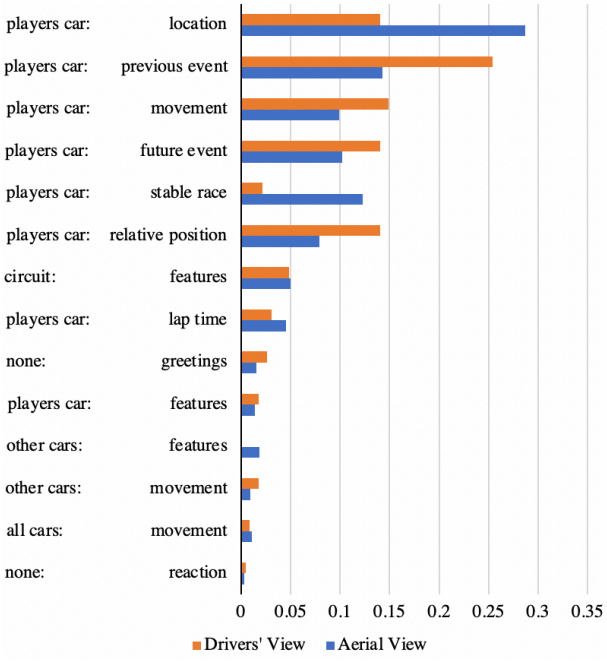

Figure 3: Distribution over utterance labels annotated from different viewpoints: driver's (the upper bars colored orange) or aerial (the lower bars colored blue).

viewpoint cannot see the entire circuit. Additionally, the proportion of *player's car:stable race* is very high for the aerial viewpoint. Because the aerial viewpoint is farther from the cars than the driver's viewpoint, the commentaries from the aerial viewpoint hardly mention slight movements of the cars; they more often say that the race is stable.

In contrast, the proportion of the *player's car:previous event* for the driver's viewpoint is higher than that of the aerial viewpoint. If the commentaries are spoken by the players themselves, they often comment on the events that just happened, e.g., *"OK, yes, the car turned successfully!"*.

The analysis above shows that the viewpoint influences the characteristics of the utterances.

4 Tasks and Models

We formulate a live commentary generation task and introduce the baseline models as shown in Figure 4. We report the performances of the baseline models for both subtasks to better understand the commentary generation task.

4.1 Task Formulation

To generate a live commentary, one needs to find multiple timepoints and generate an utterance at each timepoint. We solve this task in a sequential fashion; given the previous timepoint and its utterance, we find the next timepoint and generate its utterance, which will be solved below.

The task of timing identification is to determine the timestamp t at which an utterance should be generated. We formulate this problem as a binary classification for each second. Given the timepoint of the previously generated utterance, we iteratively classify each successive second according to whether the second is the next timepoint for generation or not. If the second is classified as positive, the model goes on to the generation step. If the second is classified as negative, the model goes on to the classification of the next second. If the model does not output positive for m seconds, the next second is forced to be positive. We set $m = 7$, which is double the average interval between two consecutive utterances.

For the classification of each second, we encode a given tuple (V, D, T). V denotes a sequence of the previous k video frames $V = (img_1, ..., img_k)$ captured every second. We set $k = 10$ in our experiments. We used `torchvision`[9] library to extract these images from videos. S denotes the structured data $D = \{D_1, ..., D_N\}$ consisting of N sets of the structured telemetry data, where each $D_n = \{val_{1,n}, ..., val_{k,n}\}$ consists of k values tracked at each of the previous k seconds. T represents the textual information, which is the previous utterance in our setting.

The task of the utterance generation is to generate a sequence of characters as an utterance, given the tuple of (V, D, T) for the given/estimated timepoint. In other words, we use the same information for both the second classification above and the utterance generation. We use a multimodal encoder-decoder architecture to generate an utterance.

4.2 Multi-modal Encoder

The models for both subtasks use the same network for encoding the input vision, structured telemetry and textual data. The encoded representation is then used in the network for subtasks. For video frames V, each video frame is converted to an image embedding by using Vision Transformer (Dosovitskiy et al., 2021)[10].

[9] https://pytorch.org/vision

[10] We used an open implementation at https://github.com/lucidrains/vit-pytorch.

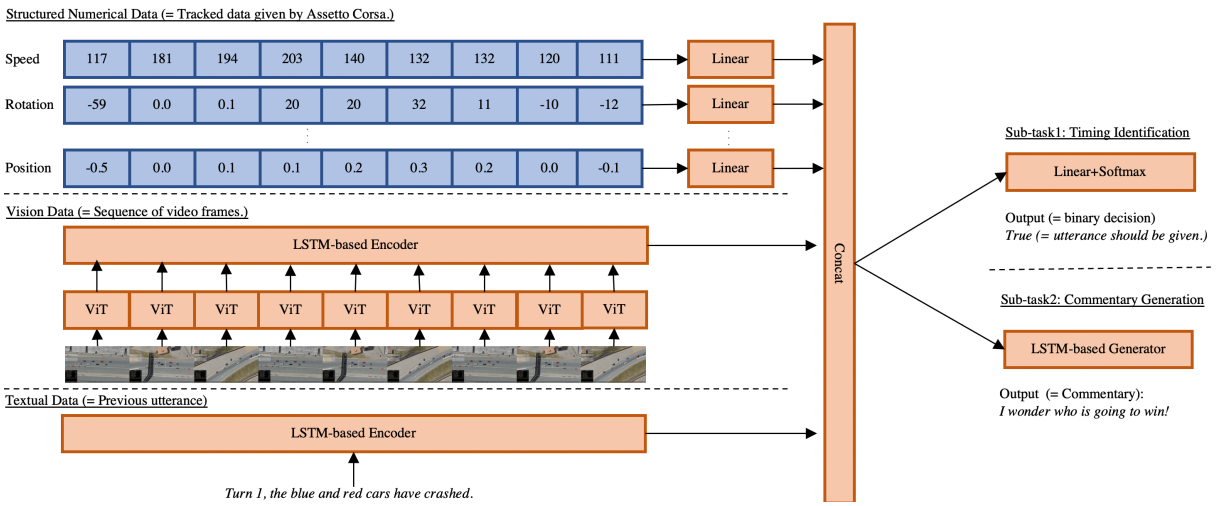

Figure 4: Baseline models for the timing identification and commentary generation tasks. Each sequence of numerical data i.e. speed, rotation, position and so on, is considered as a vector and we obtain a compressed vector. Vision information is encoded by using Vision Transformer and LSTM-based encoder. Textual information is encoded by using another LSTM-based encoder. The concatenated vector of the encoded numerical, vision and textual information is passed to the models for sub-tasks.

The image embeddings are then sequentially encoded by using a Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997): $h_{i,V} = LSTM_V(h_{i-1,V}, ViT(img_i))$, where `ViT` returns the output vector for the [CLS] token calculated by Vision Transformer. We treat the final state $h_{k,V}$ of the LSTM as the representation of V. For each sequence D_n of the structured data D tracked for the previous k seconds, we consider the sequence D_n as a vector for the n-th type of data in the structured telemetry data, and we transform it into an another representation by using a linear transformation: $d_n = ReLU(D_n W_d + b)$, where W_d is a weight matrix, b is a vector, and ReLU activates the vector. The concatenated vector of D_1 to D_N is the representation of D. For textual input, we simply embed characters in the textual input and then sequentially encode the embeddings by using another LSTM: $h_{i,T} = LSTM_T(h_{i-1,T}, emb_e(x_i))$, where emb_e returns the character embedding. We treat the final state of the LSTM as the representation for textual input T. Finally, the concatenated vector of the encoded representations of V, D, and T is passed to the networks for the sub-tasks explained next.

4.3 Timing Identification Model

For the timing identification, the encoded representation is passed to a network that consists of a linear transformation and the softmax function: $Softmax(encode([V; D; T])W_t)$, where $encode()$ returns the outputs of the encoder and W_t converts the concatenated vector to two-dimensional vector that represents the scores for the decisions to utter or not utter at this timepoint. We obtain the probability distribution over decisions by using the softmax function.

For training, we use the gold start timestamps from commentators as positive instances. We use the midpoint of the silence between consecutive utterances as negative instances. We train this classifier by using the cross-entropy loss. For testing, we classify every second after the time at which the previous utterance is given. We output the timestamp first classified as positive by our model.

4.4 Utterance Generation Model

This second subtask generates an utterance as a sequence of characters given an encoded representation V, S, and T. We use an encoder-decoder architecture with an attention mechanism, which consists of an LSTM-based decoder initialized by the representation passed from the encoder:

$$h_{j,d} = LSTM_d(h_{j-1,dec}, emb_d(y_{j-1})), \quad (1)$$

$$a_{ji} = \frac{exp(h_{j,d} W h_{i,V})}{\sum_{i=1}^{10} exp(h_{j,d} W h_{i,V})}, \quad (2)$$

$$c_j = \sum_i a_{ji} h_{i,V}, \quad (3)$$

$$o_j = Softmax([h_{j,d}; c_j] W_d), \quad (4)$$

where emb_d returns the embedding of a character,[11] c_j is a vector produced by an attention mechanism over the outputs of $LSTM_V$, and y_{j-1} is the previously generated character. W_d is a matrix that converts the concatenation $[h_{j,d}; c_j]$ to a vector of scores over the predefined vocabulary for the target utterances, and `Softmax` converts it to a probability distribution. This generator is trained by using cross-entropy loss.

5 Experiments

We conduct experiments for the two subtasks to further investigate the characteristics of the task.

5.1 Data and Parameters

We use 100 tuples of videos, commentaries, and structured data for validation, another 100 tuples for testing, and the remaining tuples for training. For Vision Transformer, we set the number of heads to six, the layer size to two, and each head is represented as a 100-dimensional vector. The parameter for the patch size is 30×30. The dropout rate was set to 0.1. Each type of telemetry data is represented as a 10-dimensional vector. We use three types of data i.e., speed, progress in a lap, steer rotation, and position on track. The dimensions of both the hidden states and input vectors to the LSTMs in encoders are set to 100. Thus, the dimension of the hidden state of the LSTM in the decoder side is 230, which is the sum of the size of the encoded images, textual information and structured data. The size of the character embeddings in the decoder is set to 100. We use separate vocabularies for the textual input and the target text. We use Adam (Kingma and Ba, 2015) with several initial learning rates ranging from 10^{-3} to 10^{-5} for optimizing parameters. We continue the training iterations until the loss in the validation dataset does not decrease for 10 epochs. We conduct the utterance generation experiments for the gold timestamps.

5.2 Timing Identification

We evaluate the models by using the average gaps in second between the gold timestamp and predicted timestamp. We propose a simple baseline that outputs the timestamp after 3.46 seconds from the end timestamp of the previous utterance. 3.46 is the average interval between two consecutive utterances as shown in Table 4. As a result, the average gap between the gold timestamps and predicted timestamps obtained from the baseline model was 3.66. When we use only structured data as input, we obtained the average gap of 3.27 seconds. Adding textual information achieved a slightly better value of 3.26, but the difference is negligible. Adding vision information improves the performance to 3.12.

[11] Note emb_d is different from emb_e.

Model	Avg. gap
baseline: average interval	3.66
struct	3.27
struct+text	3.26
struct+text+vision	**3.12**

Table 4: The average gap in seconds between the gold and predicted timestamps. Lower values are better.

Model	10^{-3}	10^{-4}	10^{-5}
struct	18.22	22.78	23.39
struct + text	18.03	23.78	23.86
struct + text + vision	17.49	22.58	**24.01**
only vision	0.30	2.74	7.46

Table 5: BLEU scores on the test dataset for the compared models trained on different learning rates. The model with the learning rate 10^{-5} achieves the best performance on the validation dataset.

5.3 Utterance Generation

We use BLEU (Papineni et al., 2002) to evaluate the baseline models for this task. The scores are shown in Table 5. The model based only on telemetry data worked well. Adding textual information improved BLEU score if the learning rate is set to lower values i.e., 10^{-4} or 10^{-5}. However, we obtained a very low BLEU score when we used only vision-based input. Adding vision information to struct+text model degraded the score if the learning rate is set to 10^{-3} or 10^{-4}. Even with a smaller learning rate, 10^{-5}, vision information did not significantly improve the performance.

6 Discussion

We list the gold and the utterances generated by the model with learning rate 10^{-4} in Table 6. Gold utterances often focus on relative position situations, as in Example 1, which requires capturing the physical relations between cars. However, as shown in the first example of a generated utterance by `data+text`, we found only a few generated utterances that mention the relative positions of

Example 1: timestamp: 00:55
`Gold`
The player is now following very close to the car ahead.
`data+text`
Now we're on Turn 10, the player is now accelerating
`data+text+vision`
I want to step on the brakes firmly here.
Example 2: timestamp: 02:04 and 02:07
`Gold`
We are now approaching the chicane on Turn 11 and 12. The player should properly use the curb and go on a straight line here, and the player showed stable race here.
`data+text`
The player should brake properly here. The player should brake properly here.

Table 6: The gold and automatically generated commentaries. Texts are translated from Japanese.

the player's car and other cars. Integrating vision information further reduces such utterances mentioning relative positions and other detailed information, and also makes utterances less specific. To generate utterances with detailed information, a model must accurately capture the information displayed in a small area of the image. However, it may be too hard for the model to, for example, capture the distance between the car driven by the player and the car just behind, or the drastic changes of speeds from the video frames shown in Figure 1, whereas telemetry data provides the useful information. From the perspective of studies on vision, methods to properly capture such features are worth exploring.

We also observed that generated commentaries contain many repetitions of the same utterance, especially utterances generated by the model with vision information. The utterances in Example 2 in Table 6 exemplifies repetitions. It should be note that the two utterances are only three seconds apart. The input to the model does not change significantly during such a short period of time, resulting in the two identical utterances. Some mechanisms to increase the diversity of utterances might alleviate this problem, which is a particular challenge in commentary generation.

We found errors in the name of a country e.g., *Nürburgring in Germany* was generated as *Nürburgring in Italy*. Such errors are also known as a common problem in other generation tasks.

7 Future Research Directions

Finally, we discuss the future directions. We noticed that evaluation is very difficult for this task. Only BLEU scores of course cannot capture the correctness because this evaluation ignores the relation between a commentary and a race represented in. However, manually checking videos, language, structured data, and generated utterances would incur a huge labor cost. An exploration into correct and efficient automatic and manual evaluation methods that consider all vision, language, and structured data should be conducted in the future. For evaluation by using BLEU, it might be helpful if we have multiple reference utterances for one timestamps. However, it is difficult to collect multiple utterances simultaneously in this task because different commentators give utterances at different timings. We leave them for an important future research direction.

Extensions of model would be considered as one of the main steps to produce better commentaries. However, more importantly, we need to explore an essential research question: "what is a good commentary?". Further analysis of the characteristics that contributes to making commentaries better need to be conducted.

8 Conclusion

In this paper, we proposed the task of generating commentaries for motor racing games. Our analysis reveals that the characteristics of utterances change over time in a race, and such changes are also caused by differences in viewpoints. They also show that combining vision, language and structured data is challenging, which worth studying in depth. For future work, exploring better methods to combine vision, language, and structured data will be a promising direction for future work. We release the data to enhance further studies on generation tasks from multimodal inputs.

Acknowledgements

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). For experiments, computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used. We thank KUNOS Simulazioni for granting us the permission to distribute our dataset of Assetto Corsa.

References

Kasumi Aoki, Akira Miyazawa, Tatsuya Ishigaki, Tatsuya Aoki, Hiroshi Noji, Keiichi Goshima, Hiroya Takamura, Yusuke Miyao, and Ichiro Kobayashi. 2021. Controlling contents in data-to-document generation with human-designed topic labels. *Computer Speech Language*, 66:101154.

Tatsuya Aoki, Akira Miyazawa, Tatsuya Ishigaki, Keiichi Goshima, Kasumi Aoki, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2018. Generating market comments referring to external resources. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 135–139.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pages 1–21.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2102–2113.

Byeong Jo Kim and Y. Choi. 2020. Automatic baseball commentary generation using deep learning. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, page 1056–1065.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*, pages 706–715.

Mitsumasa Kubo, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2013. Generating live sports updates from twitter by finding good reporters. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 527–534.

Daniel Memmert and Dominik Raabe. 2018. *Data analytics in football: Positional data collection, modelling and analysis*. Routledge.

Soichiro Murakami, Sora Tanaka, Masatsugu Hangyo, Hidetaka Kamigaito, Kotaro Funakoshi, Hiroya Takamura, and Manabu Okumura. 2021. Generating weather comments from meteorological simulations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL): Main Volume*, pages 1462–1473.

Soichiro Murakami, Akihiko Watanabe, Akira Miyazawa, Keiichi Goshima, Toshihiko Yanase, Hiroya Takamura, and Yusuke Miyao. 2017. Learning to generate market comments from stock prices. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1384.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics*, 9(0):510–527.

Michael Schaffrath. 2003. Mehr als 1:0! Bedeutung des Live-Kommentars bei Fußballubertragungen– eine explorative Fallstudie [more than 1:0! the importance of live commentary on football matches – an exploratory case study]. *Medien und Kommunikationswissenschaft*, 51.

Yasufumi Taniguchi, Yukun Feng, Hiroya Takamura, and Manabu Okumura. 2019. Generating live soccer-match commentary from play data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):7096–7103.

Yui Uehara, Tatsuya Ishigaki, Kasumi Aoki, Hiroshi Noji, Keiichi Goshima, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2020. Learning with contrastive examples for data-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING2020)*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *2015 IEEE International*

Conference on Computer Vision (ICCV), pages 4507–4515.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598.