From eSports Data to Game Commentary: Datasets, Models, and Evaluation Metrics

Zihan WANG† and Naoki YOSHINAGA††

† The University of Tokyo, 7–3–1 Hongo, Bunkyo-Ku, Tokyo 113–8656, Japan
†† Institute of Industrial Science, the University of Tokyo, 4–6–1 Komaba, Meguro-Ku, Tokyo 153–8505, Japan
E-mail: †, ††{zwang, ynaga}@tkl.iis.u-tokyo.ac.jp

Abstract Electronic sports (eSports), the sport competition using video games, has become one of the most popular sporting events now. The eSports audience needs textual commentaries for deeply understanding the games and for efficiently retrieving specific games of their interest. Therefore, in this work, we set up an eSports data-to-text generation task and tackle three fundamental problems: dataset construction, model design, and evaluation metrics. We first build a data-to-text dataset containing data records and game commentaries from the a popular eSports game, League of Legends. On this new dataset, we propose a hierarchical model to address difficulty in handling long sequences of inputs and outputs with an encoder-decoder model. The hierarchical model sets multi-level encoders for the input data. Besides, we organize and design a new set of evaluation metrics including three aspects to meet this task's goal. Experimental results on the new datasets confirm that the hierarchical structure improves the performance of the model.

Key words eSports data-to-text generation, hierarchical model, multimodal processing

1 Introduction

The benefits of gaming started to be studied and brought to the attention of the public especially in recent years [21]. eSports, as known as electronic sports or e-sports, is a form of sport competition where spectators watch players video gaming [7]. eSports contests have become one of the most important sporting events nowadays. For example, the current most popular eSports game, "League of Legends (LoL)," [5] has more than 100 million monthly active users, and its latest championship contest named "LoL's 2020 Season World Championship" is the sports event with the largest audience in 2020.

Despite the audience's great enthusiasm in watching the games, there are various inconveniences during they enjoy the games. The whole championship contest includes too many individual games. For example, LoL's 2019 Season World Championship has more than 200 individual games, and it is exhausting for most audience to watch the total set of games. The average time of the games mostly is over one hour, which is too long for most audience compared to traditional sports. Also, many audience have trouble in capturing the game's highlight points and players' important moves. This is because eSports games can usually produce visual content containing overwhelming information, and it costs people much more effort to pay attention to the details in eSports games than traditional sports games.

To fully enjoy watching eSports games and effectively learn playing skills from games played by skillful players, the game commentary is beneficial. However, human-written commentaries usually meet plenty of disadvantages. For example, human commentators have low efficiency, and most human-written commentaries lack details and are not in real-time with the game progress.

To satisfy the audience's needs and to overcome human commentators' shortcomings, here we introduce the task named data-to-text generation for eSports game commentary. The proposed task is to automatically generate a textual commentary given structured data of an eSports game. At the dataset level, this work builds a novel eSports data-to-text dataset to fill the gaps in the current researches. Besides, the goal of the eSports data-to-text generation is to maximize the outputs' correctness, fluency and game-level strategy depth, and this work will reflect these factors in the design of both method and evaluation metrics.

In summary, the overall workflow of addressing this new task includes three main processes: collecting data, deciding the model structure, and setting evaluation metrics. We will introduce the detailed approaches in the following sections. We first collect both structured game data and commentaries to build the eSports data-to-text dataset. Next, we introduce a seq2seq network model as the baseline of this work, and we discuss several modules to improve its performance. We also organize and propose a novel set of evaluation metrics considering the characteristics of eSports data.

In the experiments, we examine the model's performance through three aspects including correctness, fluency, and game-level strategy depth. Experimental results show how well our models work in this task.

Screenshot of the moment when Team Red achieves the first CHAMPION_KILL of thisgame	Data record	Human-written game commentary

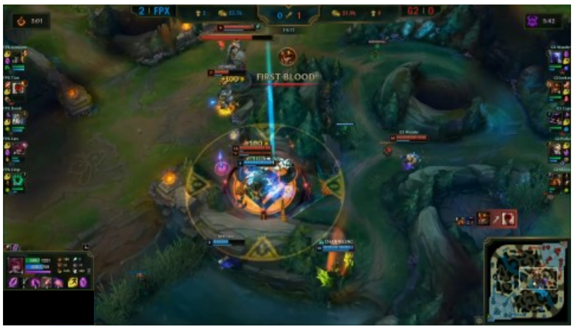
	{
	"type": "CHAMPION_KILL",
	"timestamp": 191394,
	"position": {
	"x": 4511,
	"y": 13554
	},
	"killerId": 1,
	"victimId": 6,
	"assistingParticipantIds ": [
	2,
	3
	]
	}
		lwx gets tagged the death sentence pulled back Mickey - the first blood King from Europe! but what more can they get? one that continues to chase one of the realm wolf is on cooldown

Figure 1: A "CHAMPION_KILL" event. The left column shows a screenshot of the game. The middle column shows its corresponding data record. The right column shows the human-written game commentary.

Below are the three main contributions of this paper:

- This work provides the first data-to-text dataset for eSports game.
- This work contributes to a new understanding of multimodal processing between structured data and natural language.
- At the eSports development level, this work can enable many applications to help audience understand games better and assist eSports commentators.

2 Related Work

The introduction of deep learning has improved the performance of multimodal processing and natural language generation techniques [11]. In this section, we will compare our task with other types of sports game data-to-text generation tasks and address the differences between them.

a) Game video summarization

The game video summarization is to capture key pictures [15] or to produce textual summaries [17] based on a game video to provide a quick way to overview the game's full content. These works are to explore the multimodal processing between visual and textual information, while ours is between structured data and texts. In addition to the different data modalities, the core difference between summary generation and our work is that we consider factors besides only describing the game, such as comments on the gameplay and player skills.

b) Data-to-text generation for sports games

Some researchers work on the RotoWire [22] dataset, which focuses on transcribing NBA basketball game data into textual documents. The essential difference between NBA and LoL games is that the basketball data only records certain key values (score, player number, win and lose, etc.), while eSports data provides much more details of the game. The large-scale eSports data provides both potentials and challenges for automatic processing.

c) Data-to-text generation for board games

Another popular research topic in the data-to-text generation area is concerning shogi or chess games. For such turn-based 2-player board games, the latest studies focus on generating a game commentary with individual move expressions [12, 14]. One feature which eSports and chess-like games have in common is that they can both restore a game from the game data. Nevertheless, in shogi games, the data is recorded based on states of turns, while in eSports games, it records periods of actions. This is the essential difference between turn-based games and real-time games, and it leads to the fact that the size of eSports data is significantly larger and more challenging than these board games.

3 eSports Data-to-text Task

In this work, we focus on generating game commentary from structured eSports data records. This section discusses the task settings and evaluation metrics.

3.1 Task Settings

The input of this task is data records of the eSports game LoL. The output of this task is a natural language text representing the data. Figure 1 shows an example of the data format. Detailed information of the data format is introduced later in Section 4.

3.2 Evaluation Metrics

This task's goal is to maximize the outputs' correctness, fluency and game-level strategy depth. Therefore, we have designed a specific set of evaluation metrics to follow these requirements.

a) Correctness

Correctness aims to measure whether the model produces erroneous contents and how accurate the output is to describe the game content. Therefore, we first set the word-level accuracy to count how many words in the generated output are also appearing in the reference output. Specifically, this score is calculated by: accuracy $= \frac{\text{correct word count in output}}{\text{output vocab size}}$. Also, since the events of a game are in chronological order, we should consider examining event ordering as another aspect of correctness. Approaches to basketball data-to-text generation have introduced the method analyzing how well the system orders the records discussed in the description by measuring the normalized Damerau-Levenshtein distance [2,18], and we will apply such metric in future work.

b) Fluency

Fluency aims to measure how fluent the generated output is and how easy the readers are to understand its content. Similarly to other subjects of natural language generation such as machine translation, here we calculate the BLEU score to meet the fluency evaluation [16]. BLEU is to measure how close the generated output is to the reference output by computing n-gram similarity. BLEU's output is always a number between 0 and 1. This value indicates how similar the output is to the reference output, with values closer to 1 representing more similar texts. We also calculate the perplexity to measure the fluency of the language model [10].

c) Strategy

The above two evaluation metrics can conclude essential qualities of an eSports game commentary. However, people also want the output to get a better depth of thought. In other words, besides describing the game correctly and frequently, it is better to present the game situation and players' intentions. For example, an output like "*Player 1 uses Skill A at 10 minute*" can be both correct and easy to understand while "*Player 1 uses Skill A to avoid his enemies and change his position for the upcoming fight*" is a higher-quality answer to most people. Therefore, the game-level strategy should be introduced to measure how much the model reflects these high-level contents. It is, however, difficult to design automated metrics to decide game-level strategy. Hence, we will apply human evaluation in future work for better understanding. Besides, human evaluation can help evaluate correctness and fluency.

4 Building eSports Data-to-text Dataset

The first contribution of this work is to construct an eSports data-to-text dataset. In this section, we first introduce the basic rules of the target game, League of Legends (LoL). We then discuss the methods of extracting and organizing the data from this game.

4.1 Basic Game Rules

Most eSports games such as LoL are applying the multiplayer online battle arena (MOBA) mechanism. MOBA game is a subgenre of strategy video games in which each player controls a single character with a set of unique abilities that improve over the course of a game and which contribute to the team's overall strategy [3].

In each LoL game, there are two teams compete in the certain map as shown in Figure 2. Each game team has five players, and each player controls a game character ("champion") with their unique abilities. Both teams' goal is to destroy the opponent's "nexuses (bases)" while protecting themselves. Also, the champion can kill units and destroy buildings to earn resources and improve their abilities by purchasing items and upgrading abilities.

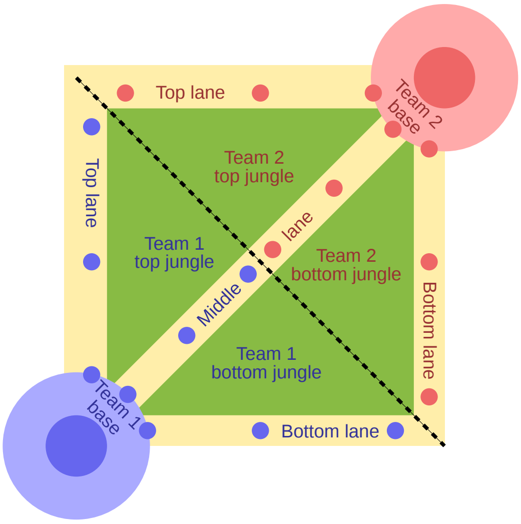


Figure 2: Game map in LoL. The yellow lines are the "lanes" where the action is focused. The blue and red dots are the "buildings" or "turrets" of two teams. The green field is the jungle area which produces jungle monster resource. The two light-colored areas are the teams' "nexuses (bases)" which encompass the blue and red corners, the structures upon which destruction results in victory. Other details including the shops, rivers and epic monsters are omitted in the map.

In conclusion, LoL games have many game terms and elements. Compared to other traditional sports games, the LoL mechanism is much more complex on game duration, play skills, complexity of rules, and many other aspects. These factors are obstacles of the data-to-text generation task.

4.2 Data Extraction

Unlike the basketball data-to-text dataset such as RotoWire, in eSports games, it is workable to record every single move of each player, including the mouse movement and keyboard click, and such records are provided by the LoL official API site [6]. In other words, from LoL data, we can strictly replay the entire corresponding game, which is impossible in traditional sports games such as basketball. This is one of the essential differences between eSports and traditional sports games.

Nevertheless, the complete eSports data have duplicate information, and this large data size is a heavy burden for storage and the following processing. For this reason, we have explored the official API documents and have chosen another data type named "event-based data frame." In this data frame, it only records the game events in individual games, where an event is defined as an update of certain game status by the game API, and Table 1 presents selected example events and their corresponding definitions. Overall, the data of each game is stored in a JSON-like file. The data record in Figure 1 presents how the example CHAMPION_KILL game event is stored in this file.

The above contents have introduced the method to extract input data. As for the output data, which is textual game commentaries in

Event type	Definition
ITEM_PURCHASED	The player purchases an item from the shop
ITEM_SOLD	The player sells an item to the shop
ITEM_UNDO	The player undoes an item related action
ITEM_DESTROYED	The item is destroyed
BUILDING_KILL	The team destroys an enemy building
CHAMPION_KILL	The player's team kills an enemy champion
SKILL_LEVEL_UP	The player upgrades its champion's ability
WARD_PLACED	The player places a ward (a special item) on the game map
WARD_KILL	The player kills a ward
ELITE_MONSTER_KILL	The player's team kills an elite monster

Table 1: Examples of the event types in the data records and their corresponding explanations.

# games	3,490
# event types	10
Avg. input length (# words)	540.47
Avg. output length (# words)	374.68

Table 2: Statistics of eSports data-to-text dataset (after pre-processing).

this work, we use contest videos' subtitles extracted from the contest video website to present comments made by human experts.

In addition, especially in this work, we choose to use game data from LoL's 2019 Season World Championship. Game data from other annual competitions will be considered for use in future work.

4.3 Data Pre-processing

a) Data Splitting

Statistical results on our collected data shows that the average length of the input structured data is over 30K words and the average length of the output commentaries is over 10K words. This length is much greater than other data-to-text generation tasks, and this leads to the problem that using too long sequences can adversely affect both memory consumption and training process. For this reason, we also prepare a modified version of this dataset, in which we split individual games into 20-40 segments based on their game duration. Our following experiments and discussions are all based on this modified dataset. We will publish both these versions for research purposes. The final dataset contains 3,490 data-commentary pairs, whose size is comparable to the RotoWire dataset [22]. Table 2 shows the detailed dataset statistics.

b) Data Linearization

We also consider the structural difference between the input data (in JSON-like format) and plain natural language texts. To reduce or prevent the impact of this structured format on training, we also linearize the input data in natural-language-like style. The following shows an example of how we linearize the JSON-like data as shown in Figure 1:

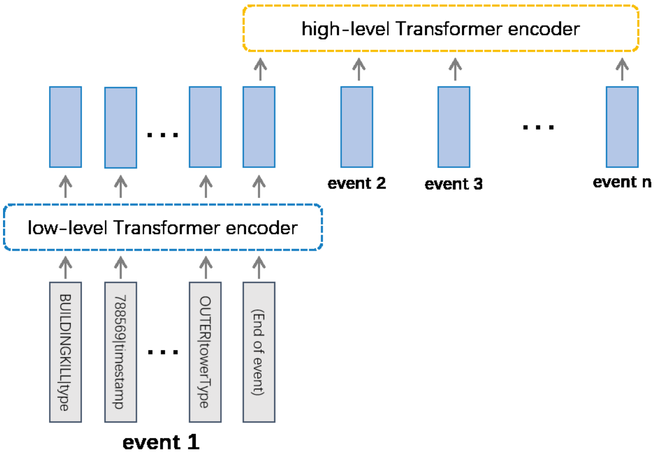


Figure 3: Hierarchical Transformer encoder. The low-level encoder works on each event independently to encode each word of the event. The high-level encoder encodes the collection of events in the game.

CHAMPION_KILL|type 191394|timestamp {x:4511,y:13554 }|position 1|killerId 6|victimId [2,3]|assistingParticipantIds

The names of the attributes are transformed as part-of-speech (POS) tags [8] and values are rewritten as natural language words. Still, we will publish two versions of the dataset for both input data formats.

5 Generating Game Commentary from eSports Data Records

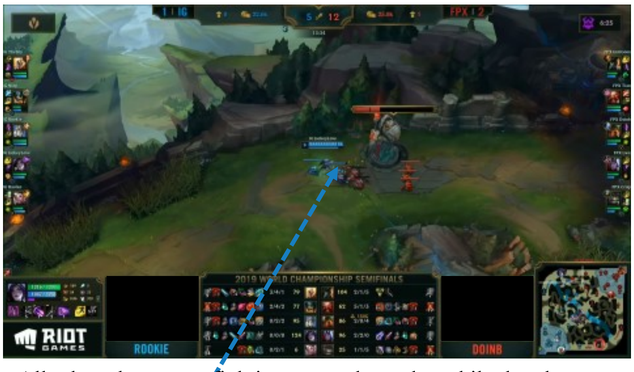
This section introduces the specific neural network architectures used in this work, including the baseline model and several advanced modules and methods.

5.1 Encoder-decoder Model

This work has applied a seq2seq network model as the baseline [19]. In specific, the model follows the encoder-decoder architecture [1]. The encoder takes game data as input and convert it into fixed length vector. The decoder then transforms it into output sequence. We use Transformer encoder and RNN-based decoder (such as LSTM [9] or GRU [4]) in this work although related approaches use RNNs as both encoder and decoder. This is because RNNs require their input to be fed sequentially, while in this task, the input is a collection of entities. RNN encodes unordered sequences by implicitly assumes an arbitrary order, and it can significantly impacts the learning performance [20]. On the other hand, since the output commentary in this task is in natural language form, we keep using LSTM decoders.

5.2 Hierarchical Encoder

As we have introduced in the dataset construction section, a very important feature of the input data is the event-based frame. Therefore, unlike normal natural language text in which every word is arranged sequentially, the game data has a distinctive 2-level structure. In specific, the game data of each individual game is composed of a



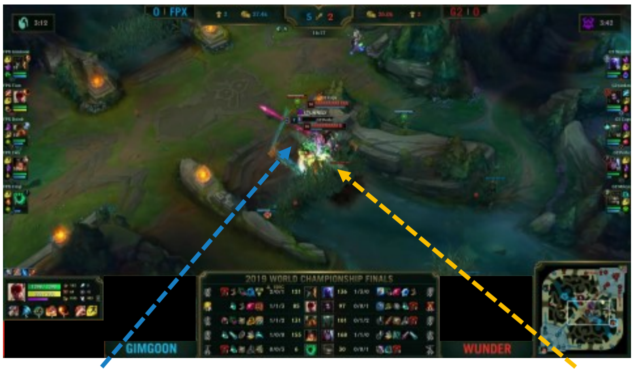
All other players are fighting somewhere else while the player Jackielove is farming alone

Ground truth: he's just farming. everyone is fighting everywhere. Jackielove is just farming for himself. everyone else rolled their characters on the PvP server, Jackie love rolled on PvE

Generated output: Jackielove is just farming a ton of these towers it's going to get caught out inside the lane turret and two towers on two here's the rift the rift Herald is going to be very important

Blue: core event Red: wrong contents

Figure 4: Visual result of selected game moment and its commentary. The left screenshot shows a BUILDING_KILL event. The core event is that the main player's champion is away from his teammates and farming to destroy the building alone. The right column shows the ground truth commentary and the generated output.



Player crisp's champion is about to be killed, and Team g2 takes this opportunity to continue their push

Ground truth: g2 will use that to try and get a Pavitt turret maybe even the full thing the dredge line doesn't

Generated output: g2 will use that to try and get a <unk> turret maybe even the full thing that feels like FBX may be trying to get some value

Blue: core event Red: wrong contents

Figure 5: Visual result of selected game moment and its commentary. The left screenshot shows a CHAMPION_KILL event. Core event of this moment is that the main player's champion is killed by his enemy team. The right column shows the ground truth commentary and the generated output.

set of events, and each event is composed of hundreds of key-value pairs (or word and POS tag pairs in the modified dataset version). This leads to the problem that the flat Transformer encoder cannot capture detailed semantics inside each data record, which may cause a large amount of information loss. For this reason, we exploit the hierarchical encoder [13, 18] architecture to replace the original Transformer encoder model.

The hierarchical encoder applies a low-level encoder and a high-level encoder, corresponding to words inside an individual event and events of a game. In other words, the low-level encoder is supposed to encode a collection of contents belonging to the same event record; and the high-level encoder is to encode the whole set of records as usual.

6 Evaluation

The dataset used in our experiments is the eSports data-to-text dataset as previously introduced with Section 4. An additional preprocessing is that we reduce the vocabulary size to 50,000 to prevent low-frequency words from affecting the training. The out-of-vocabulary words are marked as unknown words.

Method	Accuracy	BLEU	Perplexity
Encoder-decoder (baseline)	89.13	56.2	1.42
+hierarchical (proposed)	**91.30**	**56.8**	**1.32**

Table 3: Evaluation on eSports data-to-text dataset. BLEU score, word-level accuracy, and perplexity recorded.

In this work, we set the word embedding size (vector dimension) as 600. For encoder and attention, we use the hierarchical architecture as introduced in Section 5.2; for decoder, we use a LSTM network with 2 layers and 0.5 dropout. The maximum batch size for training is set to 4. Training is stopped after 30,000 steps. Starting learning rate is set to 0.001, and since the 10,000th step learning rate is decayed to 0.0005.

Current experimental results are listed in Table 3. The hierarchical model performs better than the baseline encoder-decoder model.

Figure 4 shows one game screenshot and its corresponding commentaries produced by both human commentators and the machine. The generated output is correct about the core content "the player is farming alone to take the enemy building down." However, the generated output still contains hallucinations and repetition. Also, the human commentators makes a joke about this game event (*"everyone else rolled their characters on the PvP server, Jackie love rolled on PvE"*), which is rather difficult for the machine to reproduce.

Figure 5 shows another example of a "CHAMPION_KILL" game event. Although the generated output is correct about the core content, it still contains an unknown word and makes a wrong comment on the teams' moves ("*FBX may be trying to get some value*" is a wrong comment). These are potential problems to be tackled in future work.

7 Conclusion and Future Work

This work has introduced the novel task named data-to-text generation for eSports game commentary, aiming at generating textual commentaries given structured eSports data. The automatically generated commentaries are easier to achieve higher efficiency and accuracy than human commentators. This work has finished the overall workflow containing building a novel dataset, designing the baseline network model and its improved modules, and designing corresponding evaluation metrics to meet this task's goal. In addition, we note that our approach can still produce erroneous outputs, so we are planing to finish additional modules to give the solution. We will also finish collecting human evaluation scores to get better understanding of the model.

References

[1] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. January 2015. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

[2] Eric Brill and Robert C Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 286–293, 2000.

[3] Alejandro Cannizzo and Esmitt Ramírez. Towards procedural map and character generation for the moba game genre. *Ingeniería y Ciencia*, 11(22):95–119, 2015.

[4] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[5] Simon Ferrari. From generative to conventional play: Moba and league of legends. In *DiGRA Conference*, pages 1–17, 2013.

[6] Riot Games. Riot games api. https://developer.riotgames.com/.

[7] Juho Hamari and Max Sjöblom. What is esports and why do people watch it? *Internet research*, 2017.

[8] Peter A Heeman. Pos tags and decision trees for language modeling. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

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

[10] John Horgan. From complexity to perplexity. *Scientific American*, 272(6):104–109, 1995.

[11] Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 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, 2019.

[12] Hirotaka Kameko, Shinsuke Mori, and Yoshimasa Tsuruoka. Learning a game commentary generator with grounded move expressions. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 177–184. IEEE, 2015.

[13] Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, 2019.

[14] Sanjay Modgil, Francesca Toni, Floris Bex, Ivan Bratko, Carlos I Chesnevar, Wolfgang Dvořák, Marcelo A Falappa, Xiuyi Fan, Sarah Alice Gaggl, Alejandro J García, et al. The added value of argumentation. In *Agreement technologies*, pages 357–403. Springer, 2013.

[15] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Video summarization using deep semantic features. In *Asian Conference on Computer Vision*, pages 361–377. Springer, 2016.

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[17] Ramakanth Pasunuru and Mohit Bansal. Game-based video-context dialogue. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 125–136, 2018.

[18] Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. A hierarchical model for data-to-text generation. In *European Conference on Information Retrieval*, pages 65–80. Springer, 2020.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27:3104–3112, 2014.

[20] Oriol Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets. *CoRR*, abs/1511.06391, 2016.

[21] Greg L West, Benjamin Rich Zendel, Kyoko Konishi, Jessica Benady-Chorney, Veronique D Bohbot, Isabelle Peretz, and Sylvie Belleville. Playing super mario 64 increases hippocampal grey matter in older adults. *PLoS One*, 12(12):e0187779, 2017.

[22] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Challenges in data-to-document generation. In *EMNLP*, 2017.