Vol.2021-NL-250 No.8
2021/9/28

レーシングゲーム実況生成

石垣 達也[1,a)]　トピチゴラン[1,b)]　濵園 侑美[1,2,c)]　能地 宏[1,3,d)]　小林 一郎[1,2,e)]　宮尾 祐介[1,4,f)]
高村 大也[1,g)]

概要：本稿では，新たな言語生成タスクとして，レーシングゲーム実況テキスト生成を提案する．このタスクでは，視覚情報としてレーシングゲームの録画映像，言語データとして実況発話，構造化データとして速度，ハンドル角度といった数値データを入力として扱い，視聴者が映像を視聴しながら，レースをより理解し，楽しむための実況テキストを生成する．既存の言語生成研究においては，データセットの欠如が一因となり映像，言語，構造化データの複数モダリティを同時に考慮する言語生成研究を行うことは困難であった．また，言語生成の中でも，特に実況生成においては「どのタイミングで発話するか」「何を発話するか」を最低限決定する必要があるが，例えば野球を対象とした既存研究においてはイニング間に実況を行うなど，発話タイミングがあらかじめ与えられる設定が扱われ，後者にのみ着目されてきた．本研究ではまず，映像，構造化データとそれらに対応する実況テキストが対になった大規模データセットを作成し，レース実況の特徴について分析する．分析より，実況テキストはその言語的な特徴が，時間および実況者の視点の影響を受け，変化することが分かった．さらに，実況生成タスクをタイミング同定と発話生成の 2 つのサブタスクに分割し，これらについてベースライン手法を提案する．実験より，構造化データの活用は有益である一方，視覚情報については最先端の画像エンコーダを用いたとしても，本タスクにおいて効果が限定的であり，実況生成タスクが挑戦的な課題であることが分かった．マルチモーダルな言語生成タスクのためのデータセットとして，本研究で作成したデータセットは公開する．

Racing Game Commentary Generation

Tatsuya Ishigaki[1,a)]　Goran Topić[1,b)]　Yumi Hamazono[1,2,c)]　Hiroshi Noji[1,3,d)]
Ichiro Kobayashi[1,2,e)]　Yusuke Miyao[1,4,f)]　Hiroya Takamura[1,g)]

1. はじめに

「次は最終の直線ゾーン，追い抜いていけるか？」

F1 などのモータースポーツを含むスポーツ映像には，しばしばこのような実況音声が付与され，視聴者に状況説明や実況者の主観的なコメントが伝えられる．視聴者は実況を聞きながら映像を観ることで，状況をより深く理解し，観戦をより楽しむことができる．実況は多くのスポーツ映像やビデオゲーム映像に付与され，視聴者を楽しませる重要な役割を果たし，映像そのものの価値向上も期待できる [15]．一方，実況付与には対象スポーツやイベントに関する知識や相応の話術が必要となることから，オンライン上に存在するスポーツ映像やビデオゲーム映像の多くには実況が付与されていない[*1]．

本研究では実況の一例として，図 1 に例示するような，レーシングゲーム映像に対する実況に着目する．このような実況では映像中で起こる重要なイベントについて，適切なタイミングで短時間の間に発話する必要がある．従来の

[1] 産業技術総合研究所
[2] お茶の水女子大学
[3] LeapMind Inc.
[4] 東京大学
[a)] ishigaki.tatsuya@aist.go.jp
[b)] goran.topic@aist.go.jp
[c)] hamazono-yumi@aist.go.jp
[d)] noji@aist.go.jp
[e)] koba@is.ocha.ac.jp
[f)] yusuke@is.s.u-tokyo.ac.jp
[g)] takamura.hiroya@aist.go.jp

[*1] 例えば，Twitch にアップロードされる多くの映像には実況がない (`https://www.twitch.tv`)．

© 2021 Information Processing Society of Japan



図 1 実況の例．上の実況は空撮視点，下はプレイヤー視点での実況である．

言語生成研究では主に，“何について発話するか” というプランニングの問題と，“どのように発話するか” という言語の表層化の問題に分けて研究されてきた．実況生成においては，従来から取り組まれてきたこれらの問題に加え，“いつ発話するか”，“どの程度の時間発話するか”，“どの程度詳しく発話するか” といった従来考慮されてこなかった問題に対処する必要がある．発話タイミングを同定したり，実況発話を生成するためには，映像などで表現された時系列データを考慮する必要がある．これは，例えば画像キャプション生成のような時間軸を考慮しない言語生成の設定やストーリー生成 [19], [20] といった発話タイミングがあらかじめ与えられる問題設定とは異なる．

本研究では，映像，構造化データ，テキストデータを入力とし実況テキストを生成する設定を考える*2．従来の言語生成研究において，映像や画像といった視覚情報を入力する設定は多く存在する．本研究でも映像を主な入力として考える．しかし，実況生成のために映像を正しく認識することは容易ではない．例えば，実況では車両間の細かな位置関係についてしばしば述べられ，図 1 の空撮映像のように似通ったビデオフレーム系列から車両の位置関係のわずかな違いを認識する必要がある．一方，画像キャプション生成では，「人が踊っている」などの必ずしもオブジェクト間の軽微な違いを捉える必要のないテキストの出力を想定している．このような背景から，本研究ではレースの状況をより正確に捉えるため，映像に加え，時系列構造化データも入力として用いる設定を扱う．構造化データには，レーシングカーの座標位置，速度やハンドル角度といったサーキットやレーシングカーに関する複数の数値データを平均 0.01 秒おきに計測した値が含まれている．実際の F1 レースにおいても，複数のセンサーから 300 種類以上の数値データがリアルタイムに計測されたり*3，モータースポーツ以外の分野においても，サッカー選手の位置情報を GPS で取得 [11] したりと，様々なスポーツで構造化データを活用する試みが広がっており，構造化データが存在するという仮定は非現実的な設定ではない．F1 などの実際のレースはデータ取得が困難であるのに対し，レーシングゲームであれば容易にデータを取得できる．また，ゲーム実況は大きな関心を集めており，ゲーム実況生成自体が応用上重要な研究課題である．

しかし，実況生成のための大規模なデータセットは存在しない．そこで，129,226 発話から成る大規模な実況データセットを作成した．このデータセットは 1,389 レースを人間に実際にプレイさせ，映像データおよび構造化データを取得することで作成した．さらに，これらの取得データに対し，2 つの異なる視点からの実況音声とその文字起こしを付与した．一つ目は，プレイヤー視点からの実況である．これは，プレイヤーが運転する車の背後に設置されたカメラから撮影した映像を用いて，プレイヤー自身が運転しながら実況するものである．二つ目は，プレイヤーではない第三者の視点からの実況である．これは，上空からレースを撮影した空撮映像を観ながら，プレイヤーとは異なる作業者が実況する．本稿では，まずデータセットの作

*2 実験では直前の発話もモデル内で考慮することからテキストデータも入力としている．

*3 https://aws.amazon.com/f1/

© 2021 Information Processing Society of Japan

成手法について述べ，2 つの視点による実況の違いについて分析する．分析より，実況テキストの特徴は発話する時間や視点の影響を受け，その特徴が変化することが明らかになった．

本稿では実況生成タスクをタイミング同定タスク，発話生成タスクの 2 つに分け，ベースラインモデルの性能評価実験を行う．実験より，構造化データはどちらのタスクに対しても効果的であることが分かった．一方，映像データについては既存の画像分類タスクなどで良い性能を示している画像エンコーダを用いた場合でも，実況生成タスクにおいては性能向上への寄与が限定的であることが分かった．本稿の貢献は 3 つである: (1) レーシングゲーム実況タスクを提案し，(2) 大規模な公開データセットを作成し分析，(3) ベースラインモデルの性能評価を行い，特に映像データのエンコードや構造化データとの組み合わせ部分に関して挑戦的な課題が存在することを示す．作成したデータセットは公開する[*4]．

2. 関連研究

スポーツの試合内容などを記述する言語生成課題には，リアルタイムに生成する設定 [7], [16] と，試合終了後などにまとめて記述される設定 [14] が存在する．リアルタイムに生成する設定は，音声化して映像に合わせて聞くことが想定される発話の生成と，映像とは独立して試合の状況を数分後毎に記述するテキスト速報の生成に，さらに細分化される．本研究で対象とする実況生成課題は，リアルタイム設定あり，音声化して映像に合わせて聞くことが想定される発話の生成に分類される．テキスト速報の生成としては，ツイート抽出 [10] による手法，映像データを入力としルールベースとキーワード抽出を組み合わせる手法 [7]，ニューラルネットワークによる data-to-text モデルとしてのモデル化 [16] などが存在する．リアルタイムに生成する設定においては，本来は発話タイミング同定と発話生成という 2 つの部分問題について取り組む必要があるが，これらの既存手法では発話タイミングについてはあらかじめ与えられる設定を扱っている．例えば，Kubo ら [10] はあらかじめ発話時刻を指定しており，また Kim ら [7] は発話タイミングは野球のイニング間であるとしている．本研究の対象であるモーターレースの場合，野球のようにゲームの明確な区切りが存在せず，発話タイミングは重要なコーナーに突入するタイミングやレーシングカー同士が競り合っている場面など，レース展開に応じて適切に決定する必要がある．

本研究の設定は映像に対するキャプション生成 [7] と data-to-text [16] という 2 つの問題設定の組み合わせである．映像をエンコードする標準的な手法では，映像を画像時系列として扱い，画像埋め込みをリカレントニューラルネットワークを用いて読み込む．特に，画像埋め込みに関する研究は活発に行われており，主に画像分類タスクで性能評価される [3]．実況はしばしばレースカー同士の位置関係などについて言及するため，画像埋め込み手法は小さなセグメント間に表現される細かな関係を捉える必要がある．そのため，画像分類タスクで良い性能を示す最先端の画像埋め込み手法が, 実況生成において良い性能を示すかは自明ではない．Data-to-text 生成は，日経平均などの市況データから値動きを説明するテキストの生成 [1], [2], [13], [18]，天気予報の生成 [12]，スポーツの試合の要旨 [6], [14] やテキスト速報生成 [16] など，適用分野が多岐にわたる．これらの既存研究では，時系列数値データ [13]，表形式データ [4], [14]，シミュレーション画像 [12] など単一のモダリティを考慮している．本研究では映像，テキストデータと組み合わせる複数モダリティを扱う設定となる．近年ではニューラルネットワークを用いる data-to-text モデルが活発に研究されているが，data-to-text の研究自体は自然言語処理にニューラルネットワークが適用される以前から，長らく研究されている (詳細については Gatt らによるサーベイ論文 [5] を参照されたい)．

既存の言語生成研究は主に単一の視点からのテキストを考慮している．例えば，画像キャプションは客観的なキャプションを生成する設定であり，data-to-text モデルは入力データのもっとも特徴的な部分に着目しながらテキストを生成する．Kubo ら [10] はツイートを抽出して実況として用いると，人気の高いチームへの言及が増え実況にバイアスがかかることを指摘している．実況者の視点により生成されるべき実況テキストは異なると想定されるが，既存の言語生成研究では，視点の違いによって生成テキストがどのように変化するか深く分析されてこなかった．

本研究で構築するデータセットは，実況生成のデータセットであるだけでなく，複数モダリティからの言語生成のデータセットであるという特徴も有する．また，日本語の大規模な言語生成データセットである点も特徴的である．構築したデータセットは研究目的で一般に公開する．

3. データセット

本節ではデータセットの作成方法およびその分析について述べる．

3.1 データ収集

レース映像と実況音声の収集: ゲームタイトルとしては，Assetto Corsa[*5]を採用した．理由は，Assetto Corsa のプレーデータからは，位置や速度などの構造化データを API を用いて取得することができるからである．e-sports ゲー

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

[*5] Kunos Simulazioni 社が開発するレーシングゲームである: http://www.kunos-simulazioni.com

© 2021 Information Processing Society of Japan

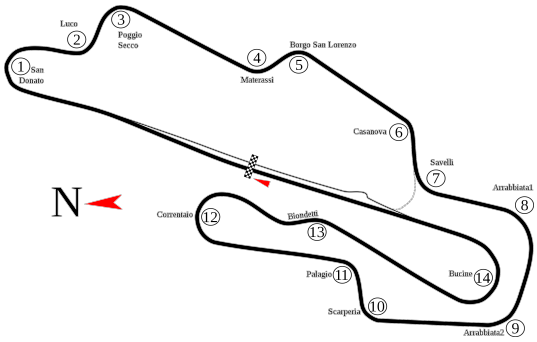


図 2 実況者に提示されたサーキットマップの例

ムを日常的にプレイする 5 人に作業をデータ作成作業を依頼した．作業者には実際にレーシングゲーム Assetto Corsa をプレイおよび音声で実況させた．また，作業者は実況についての専門家ではないが，ゲームに関連した職業についている．作業者ペアを作り 4 つのサーキットからサーキットを 1 つ選び，2 周分レースする．現実世界のレースと似た状況を再現するため，ハンドルの形をした物理コントローラーを用いてプレイした．ペアのうち一人はプレイしながら実況を行い，録音する．この作業者から見える画面は図 1 の下の画像列のように，運転する車の背後に設置されたカメラから撮影された「プレイヤー視点」の映像である．もう一人は，ゲーム内で上空を飛ぶヘリコプターからレースを撮影した「空撮視点」から実況を行う．これらの実況はすべて，日本語で行われる．

プレイヤー視点と空撮視点のいずれの視点の場合も，実況者には作業者が運転するレーシングカーに着目した実況をするよう指示した．ただし，他のレーシングカーとの位置関係など，特に言及する必要がある状況においては他のレーシングカーについて言及することも許容する．実況者には図 2 のような，サーキットについての情報が記述されたサーキットマップがあらかじめ与えられる．このマップにおいて，すべてのコーナーには番号が振られ，この番号を実況に含めることによりサーキット上での位置を参照できるよう配慮した．また，著名なコーナーには番号以外に名称が付けられている場合があり[*6]，このような名称を適宜用いることで，視聴者を楽しませる実況をするよう実況者に促した．これらのサーキットマップやコーナーの名称は Wikipedia やサーキットについて解説するウェブサイトから収集した．

実況文字起こしの収集: 映像データの収集後，クラウドソーシングサービスの Lancers[*7]を用い，149 人の作業者に文字起こしを依頼した．文字起こし作業者は実況音声をまず発話単位に分割し，文字起こしした上で発話の開始時刻と終了時刻を付与した．発話はできる限り文となるよう分割したが，言い淀みなどにより完全な文とならない発話も例外として存在する．

[*6] 例えばアメリカにあるラグナセカサーキットの第 6 コーナーは「カサノバ」と呼ばれている．
[*7] http://lancers.jp

構造化データの種類	値の例
現在の周回数 [0..]	1
レース中か否か	false
現在の周回のラップタイム (ms)	256
直前の周回のラップタイム (ms)	156164
現在のラップの進捗 [0, 1]	0.002780
ベストラップタイムとの差分	0.0
速度 (km/h)	177.693130
ハンドル角度 (rad)	-59.793526
高度，緯度，経度	(5.372770, 64.056038, 130.219971)
サーキット上の横方向位置 (L=−1, R=1)	-0.515301
理論上の最速コース取りとの距離 (m)	0.854022

表 1 構造化データの種類と例

構造化データの収集: 構造化データの収集は Assetto Corsa の提供する API を呼び出すことにより行う．この API は，構造化データとして，レースに参加するすべてのレーシングカーの速度やハンドル角度，サーキット上での座標位置などを取得することができる．収集した構造化データの一覧を表 1 に示す．この API を平均 0.01 秒おきに呼び出すことで，レース状況を表すデータを時系列数値データとして保存した．

統計量と既存データとの比較: 5 人の作業者による 1,389 レース分の実況を収集した．1,389 レースのうち 1,084 レースについてはプレイヤー視点および空撮視点の両方からの実況を収集した．残りの 305 レースについてはプレイヤー視点の実況のみ存在する[*8]．よって，合計 2,473 件の実況を収集し，これらの文字起こしデータが映像，構造化データと対応づいている．実況の録音時間は約 226 時間で，合計 129,226 発話が格納されている．映像キャプション生成の大規模な既存データセットとしては，英語の ActivityNet Captions [9] が存在するが，本データセットは日本語であり，かつより多くの発話を含む．データ中の各発話の平均長さは 2.73 秒であり，また連続する発話間の沈黙の長さは平均 3.46 秒である．英語ではいくつかの映像キャプション生成や data-to-text のための大規模データセットが存在するが，日本語の発話を格納した大規模なデータセットは貴重であり，本データセットの特徴の一つと捉えられる．その他の詳細なデータセット統計は表 2 に示す．

3.2 発話ラベルによる分析

次に発話の特徴を捉えるための分析を行う．分析のために，各発話の特徴を表現する発話ラベルを設計し，データセットの一部に人手で付与する．表 3 に示すように，1 つ

[*8] 1,084+305 動画についてはプレイヤー視点の実況，1,084 動画については空撮視点のみとなる．

© 2021 Information Processing Society of Japan

サーキット種類数	4
実況者数	5
レース数	1,389
実況数	2,473
- プレイヤー視点	1,389
- 空撮視点	1,084
実況録音時間	226:37:53
- プレイヤー視点	126:11:19
- 空撮視点	100:26:34
発話数	129,226
1 レースあたりの平均発話数	52.25
1 発話あたりの平均文字数	22.22
1 発話あたりの平均発話時間	2.73s
発話間の平均沈黙時間	3.46s

表 2 データセットの統計値

対象サブラベル	例
プレイヤーの車	ここは華麗な追い抜き，決めていく
他の車	後ろの車に抜かれましたね.
すべての車	全車今一斉にスタート.
サーキット	本日のサーキット，ここラグナセカは長いストレートで有名です.
内容サブラベル	**例**
相対位置	プレイヤー現在 *2* 位.
絶対位置	青い車今第 *2* カーブに差し掛かって，他の車もそれに続く.
タイム	プレイヤー今ゴールラインを超えて *3* 分 *15* 秒.
直前のイベント	このミスはタイムに響くかもしれない.
将来のイベント	難しいターン *15*，超えていけるか？
動き	プレイヤー，長いストレートで追い越していく.
安定したレース	全車問題なく，レースは進んでいます.
特徴	今回のレース，全車ポルシェ・マカン.
挨拶	はい，では実況を始めていきます.
反応	おぉ*!*

表 3 サブラベルの一覧．ラベルは 2 つのサブラベルの組み合わせとして定義される．

のラベルは「対象ラベル」「内容ラベル」の 2 つのサブラベルの組として定義する．1 つの発話において複数の事柄が述べられることがあるため，人手ラベリングは複数ラベルの付与を許容するマルチラベル設定となる．

1 つ目のサブラベルである「対象ラベル」は「プレイヤーの車」，「他の車」，「すべての車」，「サーキット」の 4 つから定義され，発話の言及対象を表現する．例えば，「全車一斉にスタート.」という発話であれば「すべての車」のサブラベルが付与され，「プレイヤー，今 15 コーナーを無事抜けた.」という発話であればプレイヤーの車の現在位置を対象とした発話であると考え「プレイヤーの車」を付与する．2 つ目のサブラベルである「内容ラベル」は，「(他の車との) 相対位置」「(サーキット上での) 絶対位置」「動き」「ラップタイム」といった言及対象の動きや特徴に着目した発話に付与するラベルである．例えば，前述した「プレイヤー，今 15 コーナーを無事抜けた.」という発話であれば，無事抜けたという「動き」を表すサブラベルと，「15 コーナー」という「絶対位置」を表すサブラベルが付与される．サブラベルの組み合わせが 1 つのラベルであるから，この発話には「プレイヤーの車:動き」「プレイヤーの車:絶対位置」という 2 ラベルが付与される．

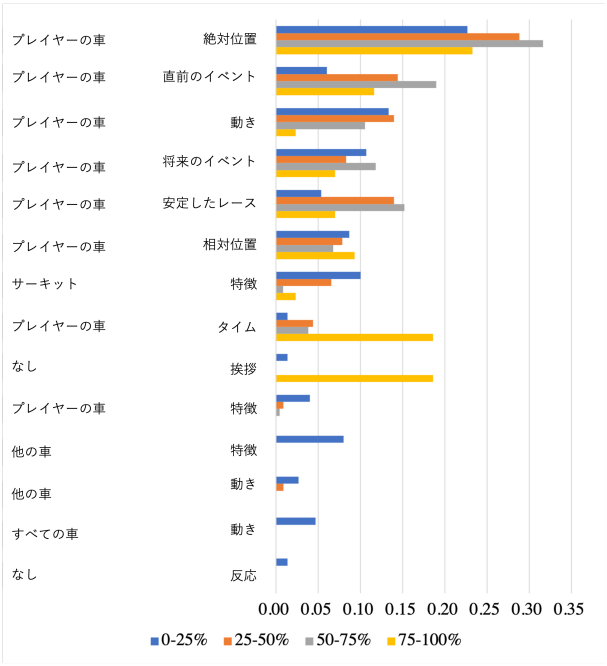


図 3 時間帯によるラベル分布の変化

無作為に抽出した 20 レースの実況に含まれる 874 発話に対し，人手でラベルを付与した．付与したラベルの出現分布を計算し，データセットに含まれる発話の特徴を捉える．実況が時系列データであるという特徴や，データセット中に異なる視点からの実況が含まれることを考慮し，(1) 時間帯によってどのように発話ラベルの分布が変化するか，(2) 実況者の視点により発話ラベルの分布が変化するか，の 2 点について特に着目し分析する．

分析 1: 時間帯により発話ラベルの分布がどのように変化するか

時間帯による発話ラベル分布の変化を調べるために，まず 1 レースを時間を手がかりに 4 分割する．すなわち，1 つ目の分割部分は録画開始時刻から時間の経過が 25%部分に達するまでとなり，以後も同様に 25%ずつ動画を 4 分割した．図 3 に各分割部分ごとの発話ラベルの分布を示す．この図において，青の線は先頭の 25%を示す．同様に，橙色，灰色，黄色はそれぞれ以後 25%ずつに分割した各区分での分布を示す．

まず，青色で示されたレース冒頭の 25%部分に着目す

© 2021 Information Processing Society of Japan

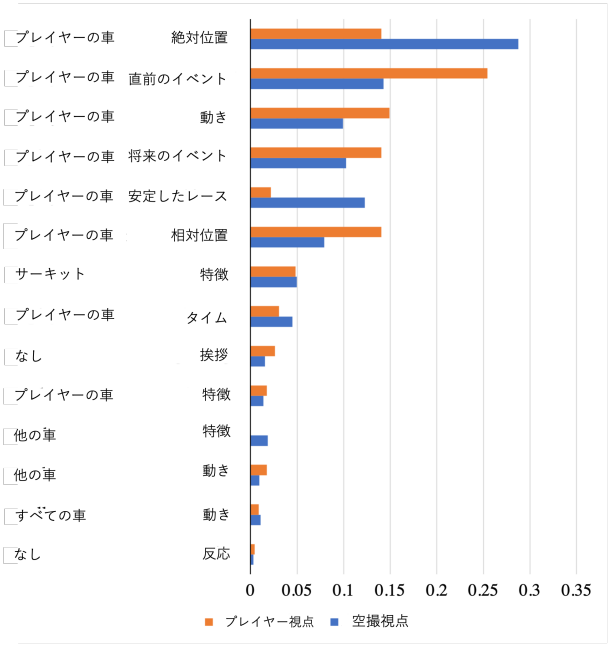


図 4 視点による発話ラベル分布の違い

る．この時間では「サーキット:特徴」「プレイヤーの車:特徴」など，「特徴」に関する言及が多いことが分かる．これは，レース開始直後にはサーキットや車の特徴について説明する発話が多いことを示している．

黄色で示された末尾の 25%の区分においては，「なし:挨拶」「プレイヤーの車:タイム」のラベル頻度が高くなっている．実際の発話を目視すると，実況者はレーシングカーがゴールラインを超えた際にタイムについて言及することが多い．また，実況の最後には「本日のレース，これにて終了です.」といった挨拶で実況を終えることが多く，そのような特徴がラベル分布に現れている．

次に，橙色と灰色で示された中間 2 つの区分における分布について分析する．この 2 つの区分は似たような分布であるが，「プレイヤーの車:動き」のような事実についての言及は，橙色で示された 25%から 50%の区分において，灰色で示された 50%から 75%の区分よりも高くなった．一方で，「直前のイベント」「将来のイベント」といったプレイを振り返ったり，将来への希望観測を述べる発話は灰色で示された 50%から 75%の区分のほうが高くなる傾向がある．この分布より，レースの前半部分では動きなどの事実に着目した言及が多く，後半に入りゴールが近づくにつれて，事実だけでなく実況者の主観的なコメントの割合が増加することが分かる．

分析 2: 実況者の視点により発話のラベル分布はどのように変化するか

プレイヤー視点と空撮視点で発話のラベル分布がどのように変化するかを分析する．図 4 に視点ごとに分けたラベル分布を示す．橙色の線がプレイヤー視点，青い線が空撮視点での分布を表す．

発話ラベルの割合に顕著な差がある箇所に着目し分析を述べる．「プレイヤーの車:(サーキット上での) 絶対位置」は青い線で示された空撮視点での割合が，プレイヤー視点の割合よりも 2 倍程度と差が大きい．これは，ヘリコプターから撮影した映像ではサーキットの広い範囲を確認することができ，容易にサーキット場での位置について言及できるためであると考えられる．それに対し，プレイヤー視点では前方向の限定された範囲の視界のみを用いるため，サーキット上での位置を把握することは難しい．そのため，絶対位置への言及が少なくなったものと考えられる．空撮視点では他にも「プレイヤーの車:安定したレース」ラベルの割合が高いのも特徴的である．空撮視点では広範囲を見ることができるものの，車両間の細かな距離を捉えるのがプレイヤー視点の映像と比較して，難しい．よって，プレイヤー視点では競り合っている状況であっても空撮視点ではそのような状況を捉えるのが難しく，レース状況が安定している旨の発話が多くなる．

一方，プレイヤー視点では「プレイヤーの車:直前のイベント」ラベルの割合が多い．これはプレイヤー自身がプレイしながら実況しているため，自身のプレイに対し反省などのコメント (例えば「オーケー，ここはうまく曲がれた！」など) を発話するケースが多いためである．このように，視点により実況の特徴が変化することが分かる．

4. 問題設定とベースライン手法

本節では実況生成タスクを定式化し，ベースライン手法を構築する．

4.1 問題設定

実況生成タスクを本研究では，(1) 発話タイミングの同定，(2) 発話生成の 2 つのサブタスクに分ける．発話タイミング同定タスクは，直前発話の発話時刻が与えられる設定で映像，構造化データ，テキストデータの 3 つ組 (V, D, T) を入力とし次の発話時刻を予測するタスクである．発話生成タスクは発話を与えるべき時刻が与えられる設定で，同様に (V, D, T) を入力としテキストを生成するタスクである．以下にこれら 2 つの問題を定式化する．

発話タイミング同定タスクは，直前の発話時刻が与えられた設定で，次の発話時刻を予測する．直前の発話が発せられた時刻が与えられると，発話タイミング同定モデルはまず直前発話の 1 秒後の時刻に対し，次の発話をすべきか否か 3 つ組の情報 (V, D, T) を入力とし二値分類する．分類器が正例と判定すればその時刻を次の発話タイミングとして出力し，分類器が負例と判定すればさらに時刻を 1 秒進め，二値分類を繰り返す．分類器が直前の発話から m 秒間分正例を出力しない場合，次の時刻では正例と判定しその時刻を発話タイミングとして出力する．実験では発話間

© 2021 Information Processing Society of Japan

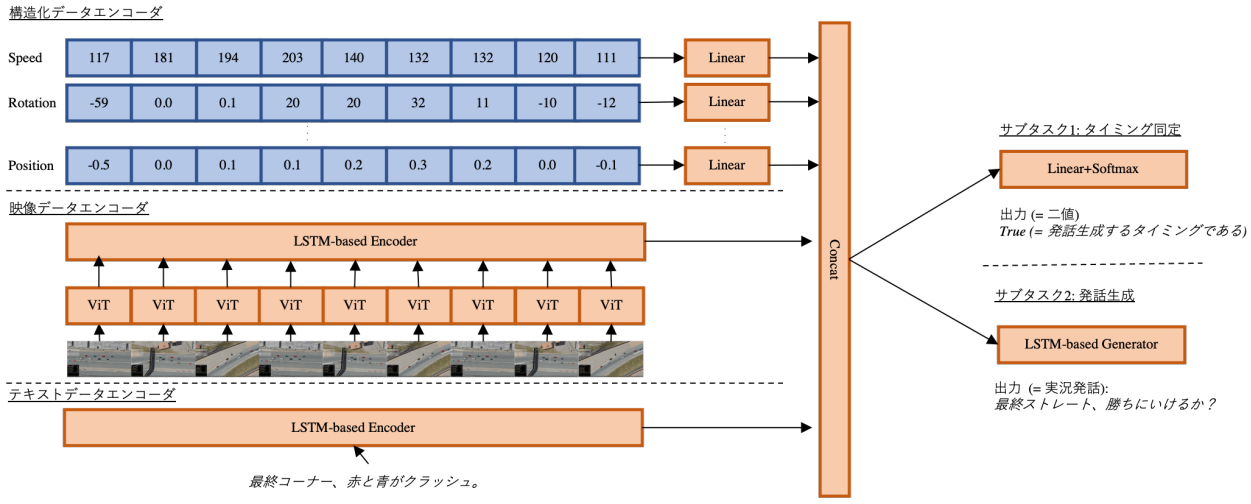


図 5 実況生成タスクのためのベースラインモデル．速度，ハンドル角度といった時系列構造化数値データは種類ごとに個別のベクトルとして表現し線形変換する．映像はビデオフレームを Vision Transformer でエンコードした後，LSTM で時系列としてさらにエンコードする．テキスト情報としては直前の発話テキストを考え，LSTM で読み込む．構造化，視覚，テキストデータの表現ベクトルをタイミング同定タスクを行う二値分類器，発話生成タスクで用いる LSTM デコーダに与える．

の平均沈黙時間の 2 倍である $m = 7$ を用いる．

発話生成タスクは映像，発話を生成すべき時刻 t が与えられた上で，3 つ組の情報 (V, D, T) を入力とし発話テキストを生成する．テキスト生成は Recurrent Neural Networks (RNN) を用いて文字を繰り返し出力し，これを終了を表す特別な文字が出力されるまで繰り返す．

4.2 ベースライン手法

本節では実験に用いたベースラインについて述べる．手法の設計においては，例えば，実況テキストのトークン分割手法やタイミング同定モデルの学習に用いる正例や負例の作成手法などにいくつかの選択肢が存在するはずである．しかし，実況生成はこれまで取り組まれたことのないタスクであり，どの手法を採用すべきかは必ずしも明らかではない．本節では，今後の精緻な手法設計のための前段階として，単純なベースラインを提案する．

マルチモーダルエンコーダ: 2 つのサブタスクにおいて，共通のマルチモーダルエンコーダを用いる．このエンコーダは映像 V，構造化データ D およびテキストデータ T をそれぞれニューラルネットワークで読み込む．$V = (img_1, ..., img_k)$ は発話タイミング同定タスクの分類対象時刻もしくは，発話生成タスクの発話生成対象時刻の k 秒前から対象時刻までのビデオフレーム系列である．ビデオフレームは 1 秒毎に `torchvision` ライブラリ[*9]を用いて取得したものを用いる．$D = \{D_1, ..., D_N\}$ は N 種類の構造化データを格納した集合である．D_1 から D_N はそれぞれ例えば速度やハンドル角度などを表現した時系列数値データに対応する．それぞれの時系列数値データ $D_n = \{val_{1,n}, ..., val_{k,n}\}$ は過去 k 秒分の時系列数値データを用いる．数値データは 1 秒おきに保存し，映像データと同様に過去 k 秒間の情報を用いる．構造化データともに過去 10 秒分，すなわち $k = 10$ を用いる．テキストデータ T には本設定では直前の発話を想定する．

V の各ビデオフレームは Vision Transformer [3][*10]で画像埋め込み表現に変換する．各埋め込み表現はその後，LSTM に入力される: $h_{i,V} = LSTM_V(h_{i-1,V}, ViT(img_i))$．ここで，`ViT` は Vision Transformer の特別なトークン [CLS] に対応する埋め込み表現を出力する．LSTM の最終状態である $h_{k,V}$ を視覚情報 V の表現ベクトルとする．構造化データ D に含まれる各数値列 D_n はベクトルとして扱い，線形変換した上で活性化させる: $d_n = ReLU(D_n W_d + b)$．ここで，W_d は各数値列を線形変換する重み行列で，b は重みベクトルである．活性化関数には $ReLU$ を用いる．D_1 から D_N までをこのネットワーク層に入力し，これらの結合ベクトルを構造化データ D の表現とする．テキストデータについては，文字ベースの LSTM でテキストデータ中の文字に対応する埋め込み表現を逐次読み込む: $h_{i,T} = LSTM_T(h_{i-1,T}, emb_e(x_i))$．ここで，$emb_e$ は対象文字の埋め込み表現を返す．LSTM の最終状態をテキストデータ T の表現とする．最後に V, D および T のエンコーダ出力の結合ベクトルを最終的なエンコーダ出力 $encode(V, D, T)$ とし，次節で述べるように各サブタスクで用いる．

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

[*10] 公開実装を用いた: `https://github.com/lucidrains/vit-pytorch`.

© 2021 Information Processing Society of Japan

タイミング同定モデル: タイミング同定モデルはエンコーダ出力を受け取り，以下の二値分類のためのネットワークを用い，発話すべきか否かについてスコアリングする: $Softmax(encode([V, D, T])W_t)$．ここで，$encode()$ はエンコーダの出力を返し，W_t を用いてエンコーダ出力を 2 次元に圧縮した上で，確率分布に変換する．2 次元ベクトルの各次元が，発話するという決定のスコア，発話しないという決定のスコアに対応する．

学習時には，データセット内で実際に発話された時間を正例とし，正解発話の終了時刻と次の正解発話の開始時刻の中点にあたる時刻を負例とした．すなわち，正例と負例は訓練データにほぼ同数存在することになる．二値分類器は交差エントロピー損失を最小化するよう学習した．

発話生成モデル: 発話生成モデルは，発話すべき時刻と 3 つ組 (V, S, T) が与えられた設定で，発話テキストを出力する．提案ベースラインでは，エンコーダ・デコーダモデルを用い，エンコーダには前述したマルチモーダルエンコーダ，デコーダには文字ベースの LSTM デコーダを用いる:

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

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

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

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

ここで，デコーダ LSTM の初期状態 $h_{j,d}$ はエンコーダ出力 $encode(V, S, T)$ で初期化する．emb_d はデコーダ側の文字埋め込み表現を返す．なお，ここでの文字埋め込み表現 emb_d はエンコーダ側の文字埋め込み表現 emb_e は異なる重み行列を用いて計算する．c_j は注意機構を用いた計算されたビデオフレーム埋め込み表現の加重平均である．y_{j-1} は直前に出力された文字の埋め込み表現で，重み行列 W_d および `Softmax` 関数を用いて，次の文字を予測する．デコーダ LSTM は，終了を表す特別なトークンを出力するまで，文字を逐次出力していく．このエンコーダ・デコーダは交差エントロピー損失を最小化するよう学習される．

5. 実験

実況生成タスクの特徴を明らかにするため，4 節で述べたベースラインモデルの性能評価を行う．

5.1 使用データとパラメータ

データセット全体から映像データ，構造化データ，実況の 3 つ組をランダムに 200 組抽出し，そのうち 100 組ずつをそれぞれ評価セット，パラメーター調整用の開発セットとして用いる．残りのデータをすべて学習用に用いる．Vision Transformer については，100 次元の隠れ状態を 6 つ持つ 2 層の Transformer に設定した．画像は 30×30 のパッチに分割し入力した．ドロップアウト率は 0.1 に設定し学習した．構造化データである速度は，直近 10 個の数値から成る．これを 10×10 行列で線形変換し ReLU 関数で活性化することで，新たな 10 次元埋め込み表現に変換した．その他の構造化表現であるレースの進捗割合およびハンドル角度についても同じ処理を行い，それぞれ 10 次元埋め込み表現に変換した．エンコーダ側の LSTM については隠れ層の次元を 100 に設定した．映像データおよびテキストを異なる LSTM で読み込み，これらと 3 つの構造化データの埋め込み表現の結合ベクトルでデコーダを初期化する．そのため，発話生成に用いるデコーダ LSTM の隠れ層は 230 次元に設定した．デコーダ側で用いる文字埋め込みは 100 次元で表現し，この文字埋め込みは直前発話を読み込むエンコーダとは異なる重み行列を用いて計算した．ネットワークの最適化には [8] を初期学習率 10^{-3}, 10^{-4} および 10^{-5} を用い，学習率ごとに複数のモデルを学習し保存した．学習は開発セットでの損失関数の値が 10 エポック向上しなくなるまで繰り返した．発話生成モデルの評価においては，正解のタイムスタンプを与える設定で生成した発話を用いて評価する．

	正解との差 [秒]
ベースライン: 平均沈黙時間	3.66
構造化データのみ	3.27
構造化データ+直前発話	3.26
構造化データ+直前発話+映像	**3.12**

表 4 実際に発話された時刻と予測時刻の差による評価．値が小さいほど性能が良い．

5.2 発話タイミング同定モデルの評価

発話タイミング同定モデルは，システムの出力した時刻と正解の時刻との差を用いて評価する．ベースラインとして，連続する 2 発話間の平均の沈黙時間である 3.66 秒を用いる．表 4 に結果を示す．なお，この表では数値が小さい方が良い性能を表す．結果，構造化データのみを用いるモデルでは 3.27 秒とベースラインの平均沈黙時間 3.66 秒よりも良い性能を示した．さらに，直前発話や映像データを考慮することで，それぞれ 3.26 秒，3.12 秒と性能向上がみられたものの，この差は小さい．このような結果から，構造化データの利用は有益であるが，映像との組み合わせについてはたとえ最先端の画像エンコーダを用いたとしても，他のモダリティのデータとの単純な組み合わせだけでは大きく性能を向上させることが難しく，エンコーダや組み合わせ手法の改善によりさらなる性能向上が見込まれる．これより，マルチモーダルな情報を考慮する発話タイミング同定タスクは挑戦的な課題であるといえる．

5.3 発話生成モデルの評価

発話生成モデルの評価には BLEU を用いる．BLEU ス

© 2021 Information Processing Society of Japan

学習率	10^{-3}	10^{-4}	10^{-5}
構造化データのみ	18.22	22.78	23.39
構造化データ + 直前発話	18.03	23.78	23.86
構造化データ + 直前発話 + 映像	17.49	22.58	**24.01**
映像のみ	0.30	2.74	7.46

表 5 異なる学習率による BLEU スコア

例 1: 発話時刻: 00:55
正解発話
*(*先頭の車に*)* 後ろピタッと付いておりますのが
プレイヤーです.
構造化データ+直前発話
さあ，ここで第１０コーナー，ここでスピードを
上げていきます.
構造化データ+直前発話+映像
さあ，ここでしっかりとブレーキをしていきます.
例 2: 発話時刻: 連続する 3 発話
正解発話
さあ，このまま次のヘアピンカーブ.
ちょっとブレーキ甘かったか.
少し膨らんでしまったが，何とか曲がることができましたね.
構造化データ+直前発話+映像
ここは，ちょっと膨らんでしまった.
ここはしっかりと減速していきたい.
ここでしっかりと減速していきたい.

表 6 出力例．カッコ内のテキストは実況には含まれないが，例示をわかりやすくするために著者らが付け加えた文言である．

コアを表 5 に示す．結果，構造化データのみを用いるモデルは映像データのみを用いるモデルよりも大幅に良い性能を示した．構造化データのみのモデルに，テキストデータ，映像データを順に追加していくと 10^{-5} の設定においては少しずつ BLEU スコアを向上させたが，差は非常に小さい．また，10^{-3} および 10^{-4} の設定においてはわずかに性能を劣化させた．この点から，Vision Transformer の学習は学習率に敏感であり，慎重な調整が必要であることがわかる．人間による正解実況は映像データのみを考慮し作成されているにも関わらず，映像のみを入力としたベースライン手法の性能は構造化データの性能よりも大きく劣っている．この点から，特に画像エンコーダや構造化データとの組み合わせ手法に関しては改善の余地があるものと思われる．今後の改善については，7 節にて議論する．

6. 議論

表 6 に正解発話と各モデルが自動生成した発話を示す．正解発話では例 1 に示すように，他の車両との相対的な位置関係に着目する発話がみられる．このような発話を生成するためには，車同士の細かな位置関係を捉える必要があり生成が難しい例である．構造化データ+直前発話モデルでは表に示す例のように，サーキット上での絶対位置について言及しており，さらに映像データを追加するとより具体性のない発話へと変化した．正解発話のように詳細な情報を含む発話を生成するためには，映像データから車の急激な減速や加速，車両間の細かな位置関係などを正確に捉える必要がある．しかし，画像分類タスク等で良い性能を示す最先端の画像エンコーダを用いたとしても，レースの実況生成のために有益な特徴が捉えるとは限らない．一方，構造化データは位置関係のような細かな情報を直接的に与えるため，性能が良いものと思われる．映像データから細かな特徴を捉えるエンコード手法や，構造化データと映像データの組み合わせ手法については，今後マルチモーダルな言語生成研究において汎用的に必要な技術であり，さらに探求すべきであろう．

さらに，特に映像データを組み合わせたモデルの出力では表 6 の例 2 に示すように，同じような発話を繰り返すエラーが散見された．例 2 に示す 3 発話のうち後半 2 発話の発話時刻は 3 秒しか離れておらず，映像データとしては非常に似たビデオフレームがが入力されている．そのため，似ているがわずかに異なるビデオフレーム系列から必要な特徴を正しく取り出すことは難しく，同一の発話を出力したものと思われる．この観点から，細かな差異を捉えるエンコーダの拡張や，発話の多様性を向上させる仕組みの導入が必要であると考える．

また，「ドイツのニュルンベルクサーキット」を「イタリアののニュルンベルクサーキット」と間違えるといったエラーも散見された．このように固有名詞がデータセット中に多く出現する事例により強く影響される問題は，他の生成研究でもすでに指摘されており，本タスクでも問題となる．サーキットの情報などを外部知識として与え，モデルに外部知識を参照させるような仕組みなどを考える必要があるだろう．

7. 今後の方向

最後に，実況生成タスクに関し今後必要となる技術に関して議論する．まず，本タスクは評価が非常に難しく，評価指標の開発は重要な方向の一つである．本稿で用いた BLEU スコアによる評価は，正解発話と出力発話の表層的な関係だけに着目し，実況生成において重要である映像や構造化データも考慮した正しさを測れない．しかしながら，BLEU を使わず人間の評価者に映像データ，構造化データ，実況を提示し評価させるのは大きなコストがかかる．また，BLEU による評価の信頼性を向上させるには，複数の参照発話が正解データに含まれることが望ましい．しかし，実況生成タスクでは複数の実況者が同じ時刻に発話することは非常に稀であり，複数の参照発話を用意すること自体が難しい．より正確で効率的な自動評価手法および人手評価手法の開発は，実況生成タスクのようなマルチ

© 2021 Information Processing Society of Japan

モーダルな設定では特に重要であろう．

前述したように映像のエンコード手法，構造化データとの組み合わせ手法などの手法開発は主となる研究の方向性である．また，映像から有用な構造化データを抽出する技術も有用だろう．しかし，良い実況モデルを作るには，そもそも「良い実況とはなにか？」という本質的な問いに対しその特徴を明らかにする必要がある．本稿では実況文字起こしの言語的な特徴に着目した分析を行ったが，映像，音声，言語を組み合わせたさらなる事例分析についても取り組む価値があるだろう．

8. おわりに

本稿では新たな生成課題としてレーシングゲーム実況の自動生成タスクを提案し，データセットを作成した．分析より，実況に含まれる発話は時系列および実況者の視点により変化することを示した．いくつかのベースラインモデルを提案し評価したところ，速度，ハンドル角度といった構造化データの情報は非常に有益である一方，映像データのエンコードは最先端の画像エンコーダを用いたとしても挑戦的な課題であることが分かった．また，構造化データと映像データを組み合わせるマルチモーダルな手法は単純な組み合わせでは大きく性能を向上させず，さらなる手法開発が必要であることを述べた．マルチモーダルな言語生成研究を加速させるため，データセットを公開する．

謝辞

本稿は INLG2021 [17] に採択された論文を基にした。この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の助成事業 (JPNP20006) の結果得られたものである。産総研の AI 橋渡しクラウド (ABCI) を利用し実験を行った。録画映像、実況音声の作成に協力いただいたデジタルハーツ社に感謝する。データセット公開の許諾について、Assetto Corsa の開発会社である KUNOS Simulazioni 社に感謝する。

参考文献

[1] Aoki, K., Miyazawa, A., Ishigaki, T., Aoki, T., Noji, H., Goshima, K., Takamura, H., Miyao, Y. and Kobayashi, I.: Controlling contents in data-to-document generation with human-designed topic labels, *Computer Speech Language*, Vol. 66, p. 101154 (2021).

[2] Aoki, T., Miyazawa, A., Ishigaki, T., Goshima, K., Aoki, K., Kobayashi, I., Takamura, H. and Miyao, Y.: Generating market comments referring to external resources, *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 135–139 (2018).

[3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, *International Conference on Learning Representations*, pp. 1–21 (2021).

[4] Gardent, C., Shimorina, A., Narayan, S. and Perez-Beltrachini, L.: Creating Training Corpora for NLG Micro-Planners, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 179–188 (2017).

[5] Gatt, A. and Krahmer, E.: Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation, *J. Artif. Int. Res.*, Vol. 61, No. 1, p. 65–170 (2018).

[6] Iso, H., Uehara, Y., Ishigaki, T., Noji, H., Aramaki, E., Kobayashi, I., Miyao, Y., Okazaki, N. and Takamura, H.: Learning to Select, Track, and Generate for Data-to-Text, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2102–2113 (2019).

[7] Kim, B. J. and Choi, Y.: Automatic baseball commentary generation using deep learning, *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (2020).

[8] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization (2014).

[9] Krishna, R., Hata, K., Ren, F., Fei-Fei, L. and Niebles, J. C.: Dense-Captioning Events in Videos, *International Conference on Computer Vision (ICCV)* (2017).

[10] Kubo, M., Sasano, R., Takamura, H. and Okumura, M.: Generating Live Sports Updates from Twitter by Finding Good Reporters, *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1, pp. 527–534 (2013).

[11] Memmert, D. and Raabe, D.: *Data analytics in football: Positional data collection, modelling and analysis*, Routledge (2018).

[12] Murakami, S., Tanaka, S., Hangyo, M., Kamigaito, H., Funakoshi, K., Takamura, H. and Okumura, M.: Generating Weather Comments from Meteorological Simulations, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL): Main Volume*, pp. 1462–1473 (2021).

[13] Murakami, S., Watanabe, A., Miyazawa, A., Goshima, K., Yanase, T., Takamura, H. and Miyao, Y.: Learning to Generate Market Comments from Stock Prices, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1374–1384 (2017).

[14] Puduppully, R. and Lapata, M.: Data-to-text Generation with Macro Planning, *Transactions of the Association for Computational Linguistics*, Vol. 9, No. 0, pp. 510–527 (2021).

[15] Schaffrath, M.: 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*, Vol. 51 (2003).

[16] Taniguchi, Y., Feng, Y., Takamura, H. and Okumura, M.: Generating Live Soccer-Match Commentary from Play Data, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 1, pp. 7096–7103 (2019).

[17] Tatsuya, I., Goran, T., Yumi, H., Hiroshi, N., Ichiro, K., Yusuke, M. and Hiroya, T.: Generating Racing Game Commentary from Vision, Language, and Structured Data, *Proceedings of The 14th International Conference on Natural Language Generation (INLG2021)* (2021).

[18] Uehara, Y., Ishigaki, T., Aoki, K., Noji, H., Goshima,

© 2021 Information Processing Society of Japan

K., Kobayashi, I., Takamura, H. and Miyao, Y.: Learning with Contrastive Examples for Data-to-Text Generation, *Proceedings of the 28th International Conference on Computational Linguistics (COLING2020)* (2020).

[19] Vinyals, O., Toshev, A., Bengio, S. and Erhan, D.: Show and tell: A neural image caption generator, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164 (2015).

[20] Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H. and Courville, A.: Describing Videos by Exploiting Temporal Structure, *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4507–4515 (2015).

© 2021 Information Processing Society of Japan