物体配置タスクにおける危険性のクロスモーダル説明生成

○飯岡雄偉，神原元就，杉浦孔明 (慶應義塾大学)

1. はじめに

日常タスクを支援するため，ユーザと自然にコミュニケーションができる生活支援ロボットの実用化は，要支援者にとって有望な解決策の一つである．特に，生活支援ロボットが動作実行前にタスクの実行に伴う危険性を予測し，ユーザに判断を仰ぐ機能は，安全性及び利便性を高める．例えば，物体を配置する際に他の物と衝突した場合，連鎖的に衝突が起こり物体が破損する危険性がある．こうした危険性について生活支援ロボットが事前に予測し，自然言語を用いてユーザに注意喚起できることは，衝突等の危険を未然に防ぐことにつながる．一方，この機能は未だに不十分である．

上記の背景から，本研究では時刻 t までの系列データを基に，時刻 $t+1$ で起こるイベントについての説明文を生成する future captioning タスクを取り扱う．例えば，生活支援ロボットがペットボトルを棚に置く際に「ロボットのアームがマグカップに接触することで，マグカップがその隣りにあるグラスに更に接触し，グラスが倒れる危険性があります」のような文を動作実行前にユーザに提示することが望ましい．しかし，本タスクは，モデルが将来のイベントを表す画像情報を利用できないという点で難しい．そのため，過去の系列データを用いた将来の画像の予測，およびキャプションの生成という 2 つの要素が求められる．

既存手法である RFCM [1] は効果的にイベント間の関係性を抽出するために，Relational Self-Attention (RSA) [2] と transformer の機構を用いている．この結果，future captioning タスクにおいて良好な結果が報告されている [1]．一方で，物体に関する中間特徴量が失われており，物体についての記述が不十分である．

そこで本論文では，物体に関する特徴量を保持する future captioning モデルとして，rec-RFCM を提案する．本手法では，時刻 t における画像特徴量についての再構成損失を導入することにより，物体に関して適切な記述を生成する．加えて，CLIP [3] で用いられている損失 (CLIP loss) を導入することにより，対応する画像と言語の特徴量の類似度を高め，適切なイベントについての説明文を生成する．以上により，物体に関して適切なキャプションの生成が期待される．

提案手法の新規性は以下に示す通りである．

- 提案手法では，物体に関する情報を保持するための再構成損失及び，画像特徴量と言語特徴量との類似度を高めるための CLIP loss を導入する．

2. 関連研究

キャプション生成分野は画像キャプション生成分野及び動画キャプション生成分野にわけられる．[4] は画像キャプション生成分野におけるサーベイ論文の一つである．[4] では，既存の画像キャプションモデルについて分類を行い，各手法を標準評価尺度によって比較している．動画キャプション生成分野のサーベイ論文の一つとして，[5] が挙げられる．[5] では，動画キャプション生成タスクにおける各手法，標準データセット，標準評価尺度のそれぞれを網羅的にまとめている．さらに，動画キャプション生成分野は，動画キャプション生成タスク，dense video captioning タスク，そして future captioning タスク等のサブタスクに分割できる．

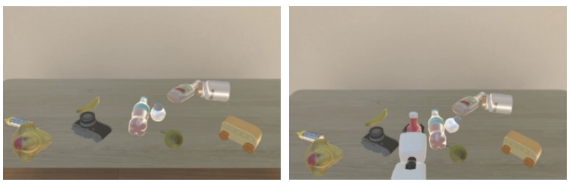


図 1 future captioning タスクの例．左図及び右図はそれぞれ時刻 t 及び時刻 $t+1$ のフレームを表す．

ABEN [6] はキャプション生成手法の一つであり，物体操作指示文付与タスクを扱う．危険予測のみを扱う手法として，PonNet [7] が挙げられる．この手法は，物体配置タスクにおいて指定された領域が危険かどうかを予測するという点で提案手法と類似している．一方，PonNet では言語生成を扱わない．

以下にキャプション生成タスクにおける標準的なデータセットを示す．YouCook2 データセット [8] は調理動画を収集したデータセットの一つである．データセットは，89 個のレシピに関する動画 2000 本を含む．それぞれの動画は，手順ごとに開始及び終了時刻，そしてその手順を説明するキャプションが付与されている．MSR-VTT データセット [9] は動画キャプション生成タスクのための広範囲な分野の動画を含む．データセットは 7K 本の動画から構成される．動画における各イベントには複数のキャプションが付与されている．それゆえに，このデータセットにはイベントとキャプションのペアが合計 200K ペア含まれる．

3. 問題設定

本論文では，future captioning タスクを扱う．本タスクの目的は， データセットにおける時刻 t までの動画フレームを入力し，時刻 $t+1$ における動画フレームの説明文を生成することである．例えば，図 1 に示すシーンにおいて，正解文は “the salt moves because robot strongly collides with the plastic bottle next to it” となる．以降，本論文では静止画である動画フレームをフレームと呼ぶこととする．本タスクにおける入出力は以下である．

- **入力**: 時刻 t までのフレーム
- **出力**: 時刻 $t+1$ におけるフレームに対する説明文

本研究は他のタスクによる事前学習を扱わないことを前提とする．これは知識転移を目的としていないためである．また，本研究では，シミュレーションによるデータを用いるが，それは以下の理由による．第一に，実機の破損を避けるためである．本論文で扱う BILA-

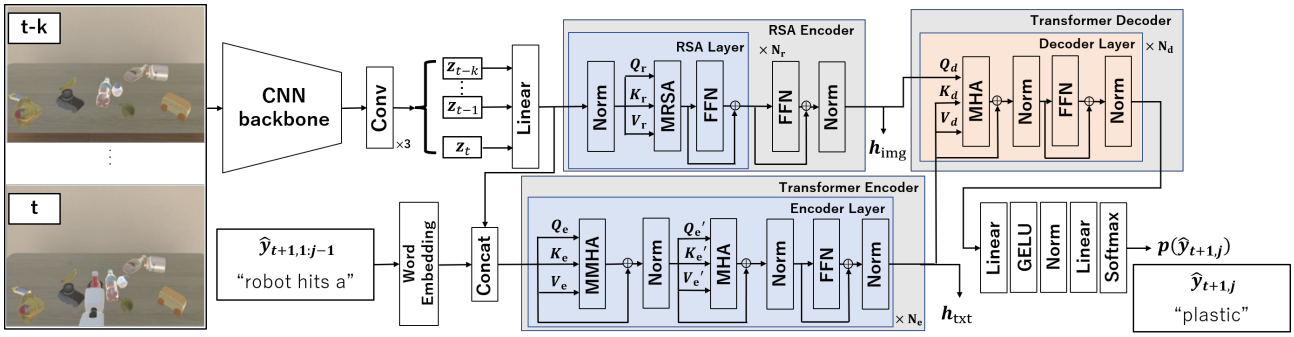


図 2 rec-RFCM のネットワーク図．Norm，FFN，MHA はそれぞれ layer normalization, feedforward network 及び Multi-Head Attention layer を示し，MRSA は multihead 化を行った RSA を表す．

Caption データセット [1] では，生活支援ロボットが日用品を配置するケースを扱う．本ケースでは，物体同士の接触が多く，実機が破損する可能性がある．加えて，データ収集にかかる時間的コストの削減のためでもある．シミュレータを用いれば，実機を用いた手作業でのデータ収集よりも低いコストで収集が可能になる．

4. 提案手法 rec-RFCM

本手法は RFCM [1] を含む既存の future captioning 手法と関連が深い．RFCM はイベントの系列データを入力とするため，系列に応じた状況の変化を学習できる．よって将来のイベントを予測し説明文を生成する future captioning タスクに適しているといえるため，RFCM を拡張した．

図 2 に提案手法 rec-RFCM のネットワーク図を示す．本手法は主に 3 つのモジュールから構成され，それぞれ RSA encoder, transformer encoder, 及び transformer decoder である．図 2 において，$\hat{\boldsymbol{y}}_{t+1,j}$, $\boldsymbol{h}_{\mathrm{img}}$, $\boldsymbol{h}_{\mathrm{txt}}$ はそれぞれ生成文における j 番目のトークン，RSA encoder で処理をして得た画像の特徴量，transformer encoder で処理をして得た文の特徴量を表す．

4.1 入力

ネットワーク入力は時刻 $t-k$ から時刻 t までのフレーム $\boldsymbol{x} = \{\boldsymbol{x}_{t-k}, ..., \boldsymbol{x}_t\} \in \mathbb{R}^{(k+1) \times 224 \times 224 \times 3}$ である．この入力 $\boldsymbol{x}$ について，ResNet-50 [10] 及び 3 層の畳み込み層から構成されるバックボーンネットワークを用いて処理をした．この処理により，各フレームについての特徴量 $\boldsymbol{h} \in \mathbb{R}^{(k+1) \times 768}$ を得た．この特徴量 $\boldsymbol{h}$ から，以下のようにして $\boldsymbol{Z}_{t-k:t} \in \mathbb{R}^{(k+1) \times d_{\mathrm{in}}}$ を得た．ただし d_{in} は各トークンのサイズとする．

$$\boldsymbol{Z}_{t-k:t} = \{\boldsymbol{z}_{t-k}; ...; \boldsymbol{z}_t\},$$
$$\boldsymbol{z}_\tau = f_z(\boldsymbol{h}_\tau, \boldsymbol{h}_t), \quad \tau = t-k, \cdots, t$$

ここで，f_z は線形変換を表す．$\boldsymbol{Z}_{t-k:t}$ は時刻 t までの履歴情報を含む．$\boldsymbol{Z}_{t-k:t}$ 及び $\boldsymbol{h}_c = \{\boldsymbol{Z}_{t-k:t}; \boldsymbol{T}\} \in \mathbb{R}^{(k+I+1) \times d_{\mathrm{in}}}$ は，それぞれ RSA encoder に入力される．$\boldsymbol{T} \in \mathbb{R}^{I \times d_{\mathrm{in}}}$ は，生成文を BERT embedder [11] により埋め込んだテキスト特徴量である．ここで，I は文の最大長を表す．訓練は teacher forcing により行う．

4.2 RSA encoder

RSA encoder は，relational kernel 及び relational context を用いて計算することによって，従来の注意機構よりも効果的にフレーム間の関係性を抽出する．本手法では，時刻 t におけるフレームの特徴量について，周辺時刻のフレームから得られる系列情報を用いてエンコードを行う．RSA encoder は n_r 層の Relational Self-Attention layer から構成され，それぞれ n_{AH} 個の並列構造を持つ MRSA により処理される．ここで，n_{AH} は attention head 数を示す．RSA layer ではまず，入力された特徴量 $\boldsymbol{h}_c$ に，[12] と同様の手順で，三角関数を用いた位置埋め込みを行う．そして，以下のように query $\boldsymbol{q}_r^{(i)}$, key $\boldsymbol{K}_r^{(i)}$, 及び value $\boldsymbol{V}_r^{(i)}$ を定義する．

$$\boldsymbol{q}_r^{(i)} = \{\boldsymbol{z}_t^{(i)} \in \mathbb{R}^{d_{\mathrm{rsa}}} | i = 1, ..., n_{\mathrm{AH}}\},$$
$$\boldsymbol{K}_r^{(i)} = \boldsymbol{V}_r^{(i)} = \{\boldsymbol{Z}_{t-k:t}^{(i)} \in \mathbb{R}^{(k+1) \times d_{\mathrm{rsa}}} | i = 1, ..., n_{\mathrm{AH}}\}$$

ここで d_{rsa} は各層のサイズを示す．続いて，以下のように basic kernel $\boldsymbol{\phi}_p^{(i)}$ 及び relational kernel $\boldsymbol{\phi}_h^{(i)}$ を獲得する．

$$\boldsymbol{\phi}_p^{(i)} = \boldsymbol{W}_p^{(i)} \boldsymbol{q}_r^{(i)\top},$$
$$\boldsymbol{\phi}_h^{(i)} = \boldsymbol{W}_h^{(i)} f_{\mathrm{flatten}}(\boldsymbol{Q}_r^{(i)} \odot \boldsymbol{K}_r^{(i)}),$$
$$\boldsymbol{Q}_r^{(i)} = \{\boldsymbol{q}_r^{(i)}; ...; \boldsymbol{q}_r^{(i)}\} \in \mathbb{R}^{(k+1) \times d_{\mathrm{rsa}}}$$

ここで，f_{flatten} は行列の平坦化を表す．また，$\boldsymbol{W}_p^{(i)}$ 及び $\boldsymbol{W}_h^{(i)}$ は訓練可能な重みである．続いて，RSA $\boldsymbol{\phi}_i \in \mathbb{R}^{d_{\mathrm{rsa}}}$ 及び relational context $\boldsymbol{\varphi}_g^{(i)} \in \mathbb{R}^{(k+1) \times d_{\mathrm{rsa}}}$ を以下のように獲得する．

$$\boldsymbol{\varphi}_g^{(i)} = \boldsymbol{V}_r^{(i)} + \boldsymbol{W}_g^{(i)} \boldsymbol{V}_r^{(i)\top} \boldsymbol{V}_r^{(i)},$$
$$\boldsymbol{\phi}_i = (\boldsymbol{\phi}_p^{(i)} + \boldsymbol{\phi}_h^{(i)})^\top \boldsymbol{\varphi}_g^{(i)}$$

ここで，$\boldsymbol{W}_g^{(i)}$ は訓練可能な重みである．$\boldsymbol{\phi}_i$ を用いて，$\boldsymbol{Z}_{t-k:t}^{(i)}$ を以下のように更新する．

$$\boldsymbol{Z}_{t-k:t}^{(i)} = \{\boldsymbol{z}_{t-k}^{(i)}; ...; \boldsymbol{z}_{t-1}^{(i)}; \boldsymbol{\phi}_i\}$$

並列に処理されたそれぞれの attention head の出力 $\boldsymbol{h}_{\mathrm{head}}^{(i)}$ を以下のように結合し，$\boldsymbol{h}_r$ を得る．

$$\boldsymbol{h}_r = \{\boldsymbol{h}_{\mathrm{head}}^{(1)}, ..., \boldsymbol{h}_{\mathrm{head}}^{(n_{\mathrm{AH}})}\},$$
$$\boldsymbol{h}_{\mathrm{head}}^{(i)} = \{\boldsymbol{z}_{t-k}^{(i)}; ...; \boldsymbol{z}_{t-1}^{(i)}; \boldsymbol{\phi}_i\}$$

RSA layer の出力 $\boldsymbol{h}_{n_{\mathrm{mrsa}}}$ $(n_{\mathrm{mrsa}} = 1, ..., n_r)$ は，$\boldsymbol{h}_r$ に Layer Norm 層，全結合層，Layer Norm 層を順に適用することにより得られる．$\boldsymbol{h}_{n_{\mathrm{mrsa}}}$ に対して，各層で同様の計算を行うことにより，最終的に，RSA encoder からは $\boldsymbol{h}_{\mathrm{img}}$ が出力される．

4.3 transformer encoder / decoder

transformer encoder は n_e 層の encoder layer から構成される．各層は transformer layer [12] を基にした構造を持つ．入力は $\boldsymbol{h}_c = \{\boldsymbol{Z}_{t-k:t}; \boldsymbol{T}\}$ である．最終層の出力に対して，FeedForwardNetwork 層及び layer normalization 層を順に適用して，$\boldsymbol{h}_{\mathrm{txt}}$ を得る．

表 1　rec-RFCM における各ハイパーパラメータ

transformer encoder	$n_e = 2$, $d_{\text{enc}} = 768$, $n_{\text{AH}} = 12$
RSA encoder	$n_r = 2$, $d_{\text{rsa}} = 768$, $n_{\text{AH}} = 12$
transformer decoder	$n_d = 5$, $d_{\text{dec}} = 768$, $n_{\text{AH}} = 12$
initial word penalty	$n_{\text{th}} = 100$, $W = 1000$
各損失の重み	$\lambda_{\text{ce}} = 0.9$, $\lambda_{\text{rec}} = 5000$, $\lambda_{\text{CLIP}} = 5$, $\lambda_{\text{iwp}} = 0.01$
最適化手法	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
学習率	1.0^{-4}
バッチサイズ	16
エポック	30

transformer decoder は n_d 層の decoder layer から構成される．各層の構造は transformer encoder における MHA 層及び FFN 層と同様のものである．query を $\boldsymbol{h}_{\text{img}}$ を基に作成し，key 及び value を $\boldsymbol{h}_{\text{txt}}$ を基に作成する．出力は $\boldsymbol{h}_{n_d} \in \mathbb{R}^{(k+I+1) \times d_{\text{dec}}}$ である．ここで，d_{dec} は各層のサイズを示す．最終的に，生成文における j トークン目の予測確率 $p(\hat{\boldsymbol{y}}_{t+1,j})$ は，$\boldsymbol{h}_{n_d}$ を全結合層，GELU 関数，layer normalization 層，全結合層，ソフトマックス関数を適用して得る．

4.4　損失関数

損失関数は以下で定義される．

$$L = \lambda_{\text{CE}} L_{\text{CE}} + \lambda_{\text{rec}} L_{\text{rec}} + \lambda_{\text{CLIP}} L_{\text{CLIP}} + \lambda_{\text{iwp}} L_{\text{iwp}}$$

上式において，λ_{CE}，λ_{rec}，λ_{CLIP}，及び λ_{iwp} はハイパーパラメータである．また，L_{CE} は交差エントロピー誤差関数を表す．L_{rec} は再構成損失を表し，画像の特徴量が失われている場合にペナルティを課す．

L_{CLIP} は CLIP [3] において用いられた損失を参考とした項である．この損失では，画像とテキストの特徴量である $\boldsymbol{h}_{\text{img}}$ と $\boldsymbol{h}_{\text{txt}}$ の内積を求め，得られる双方の類似度を高める．これによって，入力された画像特徴量に対して適切な言語特徴量を得られ，結果として適したキャプション生成が可能となると考えられる．L_{iwp} は 1 トークン目の誤りにペナルティ(initial word penalty)を課す．この損失は [1] と同様に定義される．

5.　実験設定

5.1　データセット

本論文では，BILA-Caption データセット [1] を利用した．事前処理として，動画クリップを 5 fps に変換し，各フレームを取り出した．その後，物体配置が行われる中央部分を切り抜き，リサイズを行って，最終的に $\boldsymbol{x}_t \in \mathbb{R}^{224 \times 224 \times 3}$ を得た．

5.2　ハイパーパラメータの設定

表 1 に各ハイパーパラメータの設定を示す．initial word penalty において，出現回数が n_{th} 以上の語句について考慮した．提案手法における訓練可能パラメータ数は 2.3 億であった．訓練には，メモリ 24GB 搭載の GeForce RTX 3090 および Intel Core i9-10900KF，また 64GB の RAM を使用した．モデルの訓練，および 1 サンプルあたりの推論に，それぞれ 23.5 分，および 0.47 秒かかった．early stopping の条件として，[13] に示される generalization を用いた．

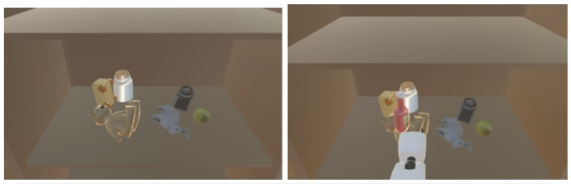
6.　実験結果

6.1　定量的結果

表 2 に定量的結果を示す．各手法について 5 回の実験を行い，その平均値と標準偏差を示した．ベースライン手法は RFCM [1] とした．

表 2　各手法における定量的結果

手法	BLEU4	METEOR	ROUGE-L	CIDEr-D
RFCM	21.74 ± 1.02	22.74 ± 0.57	41.44 ± 0.86	49.61 ± 8.02
(i)	24.55 ± 1.11	24.21 ± 0.66	**46.10 ± 0.48**	49.05 ± 3.56
(ii)	24.24 ± 0.98	24.26 ± 0.79	44.18 ± 1.03	57.32 ± 1.73
Ours	**24.82 ± 1.14**	**24.39 ± 0.73**	44.67 ± 1.13	**60.37 ± 4.31**

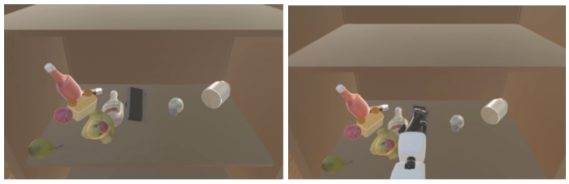


(a)

GT : “Robot bumps into the stuffed bear because robot tried to put the red bottle where it is.”

Baseline : “Robot hits the apple and the stuffed bear hard because robot tried to put the hourglass where they are.”

Ours : “Robot rubs the hand on a teddy bear because robot tried to put a red bottle.”



(b)

GT : “Robot hits the camera and a yellow toy because robot tried to put a black teapot.”

Baseline : “Robot hits a white jar because robot tried to put a round white bottle.”

Ours : “Robot bumps into the camera because robot tried to put a black teapot.”

図 3　本手法における成功例

本タスクでは，テキスト生成タスクにおける標準的尺度を用いた．具体的には，BLEU-4，METEOR，ROUGE-L，そして CIDEr-D [14] の 4 指標を用いて，人間による付与文との比較を行った．また主要尺度はキャプション生成用尺度の 1 つである CIDEr-D とした．

表 2 より，主要尺度である CIDEr-D において，提案手法及びベースライン手法はそれぞれ 60.37，および 49.61 であり，提案手法が 10.76 ポイント優れていた．

6.2　定性的結果

図 3 (a)(b) に本手法における成功例を示す．(a)(b) において，左図及び右図はそれぞれ衝突前のフレーム，及び衝突時のフレームであり，GT は正解文を表す．本実験では，左図のフレームを入力として future captioning を行った．つまり右図のフレームは入力されていない．

図 3(a) において，把持されている物体は “red bottle,” 衝突した物体は “stuffed bear” である．ベースライン手法ではそれぞれ “the hourglass”， “the apple and the stuffed bear” と誤って記述した．一方，提案手法では，それぞれ “a red bottle” および “a teddy bear” と適切に記述した．同様に図 3(b) において，把持されている物体は “black teapot”，衝突した物体は “camera” と “yelow toy” である．ベースライン手法ではそれぞれ “a round white bottle”， “a white jar” と誤って記述

表 3　本手法における失敗例のエラー分析

エラー名	エラー内容	該当サンプル数
NE	名詞に関する誤り	59
SE	深刻な記述誤り	14
OUG	記述の過不足	13
GE	文法誤り	7
合計	-	93

した．一方，提案手法では，それぞれ “a black teapot” および “the camera” と，適切に記述した．以上の結果から，提案手法はフレーム内の物体の特徴を適切に表現できたといえる．

図 4 に本手法における失敗例を示す．本例において，落下する物体は “mayonaise” であるが，本手法では “the white bottle” と誤って記述した．よって，フレームにおける解像度が低いために，物体の形状を区別する情報が不十分であったことが示唆される．

6.3　Ablation study

Ablation 条件は以下の 2 条件とした．

(i) 再構成損失が与える性能の変化を調査するため再構成損失を取り除いた．

(ii) CLIP loss の重みによる性能の差を調査するため，重みを 2 倍とした．

表 2 に ablation study の結果を示す．条件 (i) を提案手法と比較すると，主要尺度である CIDEr-D が 11.32 ポイント低下した．この結果から，本タスクにおける再構成損失の有効性が示唆された．また条件 (ii) と提案手法を比較すると，主要尺度である CIDEr-D が 3.05 ポイント向上した．この結果から，提案手法における λ_{CLIP} の値の有効性が示唆された．

6.4　エラー分析

BILA-Caption データセットのテスト集合に含まれる 100 サンプルに対する生成文について，エラー分析を行った．その結果，生成文において，深刻な記述誤りは 14 文であった．

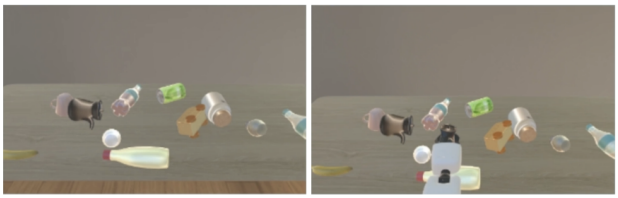
表 3 に分析結果を示す．以下が表で用いられている各エラー名の定義である．NE は生成文において，名詞が誤っていることを指す．例として，ロボットがテディベアに衝突するフレームに対して，“robot hits a pear” と記述したものが挙げられる．SE は生成文において，名詞及び動詞がともに誤っていることを指す．例として，砂時計が落ちるフレームに対して，“robot pushes the green can” と記述したものが挙げられる．OUG はフレームに対して不十分，もしくは過剰な生成をしていることを指す．例として，ロボットがテディベアとペットボトルの 2 つに衝突するフレームに対して，“robot hits the PET bottle” と記述したものが挙げられる．GE は生成文において，文法上の誤りを含むことを指す．例として，“there was it in the robot robot orbit” と記述したものが挙げられる．表 3 より，主要なエラー原因は名詞に関する誤りだと言える．原因として，入力するフレームの解像度が低いために，物体の判別を誤ってしまったと考えられる．

7.　結論

本論文では，時刻 t までのフレームによって時刻 $t+1$ のフレームのキャプションを生成する，future captioning タスクを扱った．本研究の主要な貢献は以下である．

- 再構成損失及び CLIP loss を導入した future captioning モデル，rec-RFCM を提案した．
- 提案手法の性能は，BILA-Caption データセットにおける future captioning タスクについて，ベースライン手法の性能を上回った．

将来研究として，時刻 $t+l(l>1)$ におけるフレームについてのキャプション生成が考えられる．



GT：“The mayonnaise falls from the desk because robot hits it like crushing it.”

Ours：“The white bottle in the foreground falls off the shelf because the arm hit.”

図 4　本手法における名詞に関する誤りの例

謝辞

本研究の一部は，JSPS 科研費 20H04269，JST ムーンショット，NEDO の助成を受けて実施されたものである．

参　考　文　献

[1] M. Kambara and K. Sugiura, “Relational Future Captioning Model for Explaining likely Collisions in Daily Tasks,” ICIP, 2022 to appear.

[2] M. Kim, H. Kwon, C. Wang, S. Kwak, and M. Cho, “Relational Self-Attention: What’s Missing in Attention for Video Understanding,” NeurIPS, 2021.

[3] A. Radford, J. Kim, C. Hallacy, G. Goh, S. Agarwal, G. Sastry, et al., “Learning Transferable Visual Models From Natural Language Supervision,” ICML, 2021.

[4] Z. Hossain, F. Sohel, M. Shiratuddin, et al., “A Comprehensive Survey of Deep Learning for Image Captioning,” ACM CSUR, vol.51, no.6, pp.1–36, 2019.

[5] N. Aafaq, A. Mian, W. Liu, et al., “Video Description: A Survey of Methods, Datasets, and Evaluation Metrics,” ACM CSUR, vol.52, no.6, pp.1–37, 2019.

[6] T. Ogura, et al., “Alleviating the Burden of Labeling: Sentence Generation by Attention Branch Encoder-Decoder Network,” IEEE RA-L, vol.5, pp.5945–5952, 2020.

[7] A. Magassouba, K. Sugiura, A. Nakayama, T. Hirakawa, et al., “Predicting and Attending to Damaging Collisions for Placing Everyday Objects in Photo-realistic Simulations,” Advanced Robotics, vol.35, no.12, pp.1–13, 2021.

[8] L. Zhou, C. Xu, et al., “Towards Automatic Learning of Procedures from Web Instructional Videos,” AAAI, 2018.

[9] J. Xu, T. Mei, T. Yao, and Y. Rui, “MSR-VTT: A Large Video Description Dataset for Bridging Video and Language,” CVPR, pp.5288–5296, 2016.

[10] K. He, X. Zhang, S. Ren, et al., “Deep Residual Learning for Image Recognition,” CVPR, pp.770–778, 2016.

[11] J. Devlin, M. Chang, K. Lee, et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” NAACL-HLT, pp.4171–4186, 2019.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” NeurIPS, pp.5998–6008, 2017.

[13] L. Prechelt, “Automatic Early Stopping Using Cross Validation: Quantifying the Criteria,” Neural Networks, vol.11, no.4, pp.761–767, 1998.

[14] R. Vedantam, et al., “CIDEr: Consensus-based Image Description Evaluation,” CVPR, pp.4566–4575, 2015.