物体操作タスクにおけるSwitching Funnel UNITERによる対象物体および配置目標に関する指示文理解

○是方諒介，吉田悠，石川慎太朗，杉浦孔明（慶應義塾大学）

1. はじめに

高齢化が進行する現代社会において，日常生活における介助支援の需要は高まっている．その結果，在宅介助者不足が社会問題となっており，一つの解決策として被介助者を物理的に支援することが可能な生活支援ロボットに注目が集まっている．利便性向上のためには自然言語を用いた生活支援ロボットとの対話が望まれるが，人間からの自然言語による指示をロボットが理解する能力についてはいまだ不十分である．

本研究では，物体の把持および運搬に関する物体操作指示文を生活支援ロボットが理解するための手法の構築を目的とする．具体的には，“Move the frying pan to the white table.” という指示文が与えられたときに，ロボットがフライパンを対象物体として，白い机を配置目標として認識することが望ましい．

しかし，人間の発する指示はしばしば曖昧であり，対象となる物体やその配置目標をロボットが特定することは困難である．実際に，物体操作を含む Vision-Language Navigation (VLN) における標準ベンチマークである ALFRED [1] では，人間の精度は 91.0%と報告されている一方，最先端の手法 (e.g. FILM [2]) では 30%以下しか達成できていない．

Funnel UNITER [3] は，物体操作指示文が把持対象とする物体を特定する Multimodal Language Understanding for Fetching Instruction (MLU-FI) において，Funnel Transformer [4] を導入することで計算コストを削減しつつ高い精度を達成したモデルである．しかし，この手法に配置目標候補の入力を増やすことで本研究で扱うタスクに拡張した場合，画像中に多数存在する対象物体候補と配置目標候補に関するすべての組合せについて推論を行うため多くの推論回数を要する．例えば対象物体候補および配置目標候補がそれぞれ 100 個存在する場合，もっとも尤もらしい組の探索に合計 10000 回の推論が必要となる．1 回の推論時間を 0.004 秒と仮定すると，ロボットの判断に要する時間が 40 秒と見込まれ，リアルタイム性で実用面に問題がある．

本研究では，対象物体候補および配置目標候補に関する予測を個別に行う方法でタスクを解くことが可能な Switching Funnel UNITER を提案する．これにより，対象物体候補が M 個，配置目標候補が N 個存在する状況で対象物体と配置目標の組を探索するために必要な推論回数を $O(M \times N)$ から $O(M+N)$ に削減することが可能となる．既存手法と異なる点は，Switcher およびマルチタスク学習を導入することで，単一モデルで対象物体候補および配置目標候補のどちらも入力として扱い，効率的に推論することが可能な点である．

本研究の新規性を以下に示す．

- Funnel UNITER に Switcher およびマルチタスク学習を導入することで，単一モデルで対象物体候補および配置目標候補のどちらも推論可能にする．
- 対象物体候補および配置目標候補に関する予測を個別に行うことで，少ない推論回数での対象物体および配置目標の探索を可能にする．

対象物体　　配置目標

図 1 DREC のシーン例．“Move the frying pan to the white table.” という指示文が与えられ，赤色の矩形領域を対象物体，橙色を配置目標と特定する．

2. 関連研究

マルチモーダル言語処理分野のサーベイ論文として，[5] が挙げられる．[5] は，画像および言語を統合した 10 個の代表的なタスクにおいて，問題設定，手法，既存データセット，評価指標に関して議論し，それらの結果を比較している．

MTCM-AB [6] は，MTCM [7] を ABN [8] によって拡張した MLU-FI モデルである．attention branch によってマルチモーダルな注意機構を実現し，画像中の物体の attention map を生成する．Target-dependent UNITER [9] および Funnel UNITER [3] は，UNITER [10] を対象物体候補の画像および位置情報を扱うように拡張した MLU-FI モデルである．

3. 問題設定

本論文で扱うタスクは Dual Referring Expression Comprehension (DREC) である．DREC とは，物体検出により獲得した各物体および配置先の中から，物体操作に関する参照表現を含む指示文の対象物体および配置目標の両方を特定するというタスクである．図 1 に，DREC の例を示す．

本タスクにおける入出力を以下のように定義する．

- **入力:** 物体操作に関する指示文，対象物体候補の領域，配置目標候補の領域，画像中の各物体および配置先の領域
- **出力:** 対象物体候補および配置目標候補が，それぞれ対象物体，配置目標に一致する確率の予測値

対象物体候補および配置目標候補が対象物体および配置目標にともに一致するならば 1 を，そうでなければ 0 を出力することが望ましい．

本論文で使用する用語を以下のように定義する．

- **対象物体:** 指示文が対象としている物体
- **対象物体候補:** 対象物体であるか判定する物体
- **配置目標:** 指示文が目標としている配置先
- **配置目標候補:** 配置目標であるか判定する配置先

その他の設定は，MLU-FI [3] と同様である．

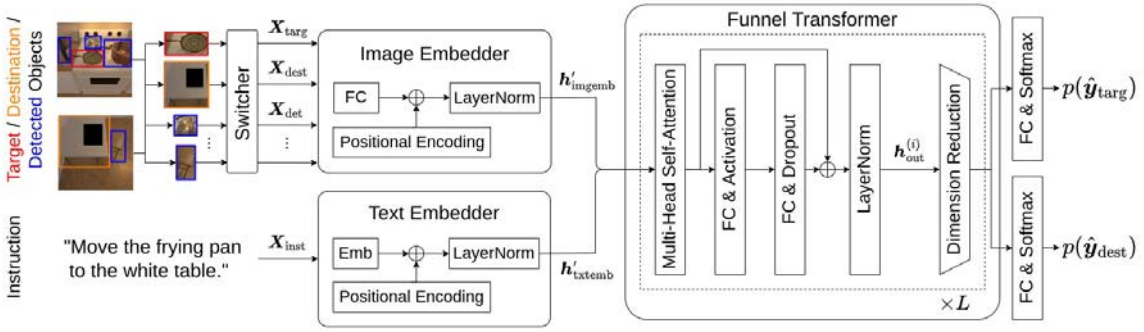

図 2 提案手法のネットワーク構造．Target Object は対象物体候補の領域，Destination Object は配置目標候補の領域，Detected Object は画像中の各物体または配置先の領域，Instruction は指示文を示す．角の丸い矢印は連結である．

4. 提案手法

図 2 にネットワークの構造を示す．モデル全体は，Switcher，Image Embedder，Text Embedder，Funnel Transformer という 4 つのモジュールから構成される．

4.1 入力

入力を $\boldsymbol{x} = \{\boldsymbol{X}_{\text{targ}}, \boldsymbol{X}_{\text{dest}}, \boldsymbol{X}_{\text{det}}, \boldsymbol{X}_{\text{inst}}\}$ と定義する．

$$\boldsymbol{X}_{\text{targ}} = \{\boldsymbol{x}_{\text{targ}}, \boldsymbol{x}_{\text{targloc}}\} \tag{1}$$

$$\boldsymbol{X}_{\text{dest}} = \{\boldsymbol{x}_{\text{dest}}, \boldsymbol{x}_{\text{destloc}}\} \tag{2}$$

$$\boldsymbol{X}_{\text{det}} = \{(\boldsymbol{x}_{\text{det}}^{(i)}, \boldsymbol{x}_{\text{detloc}}^{(i)}) \mid i = 1, \ldots, N\} \tag{3}$$

$$\boldsymbol{X}_{\text{inst}} = \{\boldsymbol{x}_{\text{inst}}, \boldsymbol{x}_{\text{pos}}\} \tag{4}$$

ここに，$\boldsymbol{x}_{\text{targ}}$ は対象物体候補の領域，$\boldsymbol{x}_{\text{dest}}$ は配置目標候補の領域，$\boldsymbol{x}_{\text{det}}^{(i)}$ は画像中の各物体または配置先の領域，$\boldsymbol{x}_{\text{inst}}$ は指示文を表し，$\boldsymbol{x}_{\text{targloc}}$ は対象物体候補の領域位置，$\boldsymbol{x}_{\text{destloc}}$ は配置目標候補の領域位置，$\boldsymbol{x}_{\text{detloc}}^{(i)}$ は画像中の各物体または配置先の領域位置，$\boldsymbol{x}_{\text{pos}}$ は指示文中の各単語の位置を表す．また，N は Faster R-CNN [11] により検出された画像中の領域数である．

$\boldsymbol{x}_{\text{targ}}$，$\boldsymbol{x}_{\text{dest}}$，$\boldsymbol{x}_{\text{det}}^{(i)}$ については，Faster R-CNN のバックボーンネットワークである ResNet50 の fc6 層の出力を，画像領域に関する 1024 次元の特徴量として抽出した．$\boldsymbol{x}_{\text{targloc}}$，$\boldsymbol{x}_{\text{destloc}}$，$\boldsymbol{x}_{\text{detloc}}^{(i)}$ については，入力画像の幅および高さをそれぞれ W，H，矩形領域の左上および右下の頂点座標をそれぞれ (x_1, y_1)，(x_2, y_2)，矩形領域の幅および高さをそれぞれ w，h として 7 次元のベクトル $[\frac{x_1}{W}, \frac{y_1}{H}, \frac{x_2}{W}, \frac{y_2}{H}, \frac{w}{W}, \frac{h}{H}, \frac{w \times h}{W \times H}]$ を得た．また，指示文に対して WordPiece によるトークン化を行うことで得た token id を $\boldsymbol{x}_{\text{inst}}$，指示文中の単語の位置を $\boldsymbol{x}_{\text{pos}}$ とし，それぞれ one-hot ベクトル集合で表現した．

4.2 Switcher

Switcher では，対象物体候補および配置目標候補のどちらに関して予測を行うかに応じて入力を切り替える処理を行う．単純な手法では対象物体候補および配置目標候補がともに対象物体および配置目標に一致するかを同時に推論する一方，本手法ではおのおのについて個別に推論を行う．これにより，対象物体候補が M 個，配置目標候補が N 個存在する状況で対象物体と配置目標の組を探索するために必要な推論回数を $O(M \times N)$ から $O(M + N)$ に削減することが可能となる．

ここで，対象物体候補および配置目標候補について予測を行うことをそれぞれ target mode，destination mode とする．Switcher への入力は $\boldsymbol{x}_{\text{targ}}$，$\boldsymbol{x}_{\text{targloc}}$，$\boldsymbol{x}_{\text{dest}}$，$\boldsymbol{x}_{\text{destloc}}$，$\boldsymbol{x}_{\text{det}}^{(i)}$，$\boldsymbol{x}_{\text{detloc}}^{(i)}$ から構成され，これらの値をそれぞれ変更することでモードの切り替えを行う．

まず $\boldsymbol{x}_{\text{targ}}$，$\boldsymbol{x}_{\text{targloc}}$，$\boldsymbol{x}_{\text{dest}}$，$\boldsymbol{x}_{\text{destloc}}$ について，各モードにおいて不要な入力を 0 埋めして出力とする．つまり，0 埋めが条件付けとして機能する．以上の処理を $\boldsymbol{x}_{\text{targ}}$ と $\boldsymbol{x}_{\text{dest}}$ について以下に示し，$\boldsymbol{x}_{\text{targloc}}$ と $\boldsymbol{x}_{\text{destloc}}$ についても同様の処理を行う．

$$(\boldsymbol{x}_{\text{targ}}, \boldsymbol{x}_{\text{dest}}) = \begin{cases} (\boldsymbol{x}_{\text{targ}}, \boldsymbol{0}) & \text{if } m = 0 \\ (\boldsymbol{0}, \boldsymbol{x}_{\text{dest}}) & \text{if } m = 1 \end{cases} \tag{5}$$

ここで，$m = 0$ は target mode，$m = 1$ は destination mode を示す．次に $\boldsymbol{x}_{\text{det}}^{(i)}$，$\boldsymbol{x}_{\text{detloc}}^{(i)}$ について，target mode では画像中の各物体の領域およびその位置，destination mode では各配置先の領域およびその位置を出力とする．

4.3 Image Embedder および Text Embedder

Image Embedder では，対象物体候補，配置目標候補および，画像中の各物体または配置先の領域に対する埋め込み処理を行う．入力は，$\boldsymbol{x}_{\text{targ}}$，$\boldsymbol{x}_{\text{targloc}}$，$\boldsymbol{x}_{\text{dest}}$，$\boldsymbol{x}_{\text{destloc}}$，$\boldsymbol{x}_{\text{det}}^{(i)}$，$\boldsymbol{x}_{\text{detloc}}^{(i)}$ から構成される．まず，$\boldsymbol{x}_{\text{targ}}$，$\boldsymbol{x}_{\text{targloc}}$ について以下の式により $\boldsymbol{h}'_{\text{targ}}$ を得る．

$$\boldsymbol{h}'_{\text{targ}} = f_{\text{LN}}(f_{\text{FC}}(\boldsymbol{x}_{\text{targ}}) + f_{\text{FC}}(\boldsymbol{x}_{\text{targloc}})) \tag{6}$$

ここで，f_{LN}，f_{FC} はそれぞれ正規化層，全結合層を示す．続いて，$\boldsymbol{x}_{\text{dest}}$，$\boldsymbol{x}_{\text{destloc}}$，$\boldsymbol{x}_{\text{det}}^{(i)}$，$\boldsymbol{x}_{\text{detloc}}^{(i)}$ についても同様に，それぞれ $\boldsymbol{h}'_{\text{dest}}$，$\boldsymbol{h}'_{\text{det}}{}^{(i)}$ を得る．最後に，$\boldsymbol{h}'_{\text{targ}}$，$\boldsymbol{h}'_{\text{dest}}$，$\boldsymbol{h}'_{\text{det}}{}^{(i)}$ を連結して出力 $\boldsymbol{h}'_{\text{imgemb}}$ を得る．

Text Embedder では，指示文に対する埋め込みを行う．入力は $\boldsymbol{x}_{\text{inst}}$，$\boldsymbol{x}_{\text{pos}}$ から構成され，以下のように出力 $\boldsymbol{h}'_{\text{txtemb}}$ を得る．ここで，$\boldsymbol{W}_{\text{inst}}$，$\boldsymbol{W}_{\text{pos}}$ は重みである．

$$\boldsymbol{h}'_{\text{txtemb}} = f_{\text{LN}}(\boldsymbol{W}_{\text{inst}}\boldsymbol{x}_{\text{inst}} + \boldsymbol{W}_{\text{pos}}\boldsymbol{x}_{\text{pos}}) \tag{7}$$

4.4 Funnel Transformer

本モジュールでは，L 層の Funnel Transformer [4] によりモデルの最終的な予測確率を得る．1 層目の入力は，$\boldsymbol{h}_{\text{in}}^{(1)} = \{\boldsymbol{h}'_{\text{imgemb}}, \boldsymbol{h}'_{\text{txtemb}}\}$ とする．

まず，transformer [12] に基づいて query，key，value を計算し，attention スコア $\boldsymbol{S}_{\text{attn}}^{(i)}$ を得る．ここで，i は Funnel Transformer の層に関するインデックスを示す．query，key，value を画像中の各対象物体候補および配置目標候補，指示文における各単語に関する特徴量を連結したものにすることで，これらの関係性を獲得することができる．

次に，Funnel UNITER [3] と同様に $\boldsymbol{S}_{\text{attn}}^{(i)}$ から i 層目の出力 $\boldsymbol{h}_{\text{out}}^{(i)}$ を得る．$i+1$ 層目における入力 $\boldsymbol{h}_{\text{in}}^{(i+1)}$ は，$\boldsymbol{h}_{\text{out}}^{(i)}$ に対して max pooling を用いて次元数を削減したものとする．i 層目の場合と同様に $\boldsymbol{h}_{\text{in}}^{(i+1)}$ を処理していく流れを L 層目まで繰り返すことで，Funnel Transformer の出力 $\boldsymbol{h}'_{\text{out}}$ を得る．

最後に，$\boldsymbol{h}'_{\text{out}}$ を 2 つの全結合層および softmax 層に入力して分岐させることによって，それぞれ対象物体候補および配置目標候補に関する予測確率を示す $p(\hat{\boldsymbol{y}}_{\text{targ}})$ および $p(\hat{\boldsymbol{y}}_{\text{dest}})$ を得る．target mode においては $p(\hat{\boldsymbol{y}}_{\text{targ}})$ を，destination mode においては $p(\hat{\boldsymbol{y}}_{\text{dest}})$ をモデル全体の最終的な出力とみなす．

マルチタスク学習の損失関数 $\mathcal{L}$ は以下とする．

$$\mathcal{L} = \lambda_{\text{targ}}\mathcal{L}_{\text{targ}} + \lambda_{\text{dest}}\mathcal{L}_{\text{dest}} \tag{8}$$

ここで，$\mathcal{L}_{\text{targ}}$，$\mathcal{L}_{\text{dest}}$ は各モードにおける交差エントロピー誤差，λ_{targ}，λ_{dest} は各タスクの重み係数を示す．なお，target mode においては $\mathcal{L}_{\text{dest}} = 0$，destination mode においては $\mathcal{L}_{\text{targ}} = 0$ として扱う．

5. 実験設定

5.1 ALFRED-fc

本研究では，ALFRED [1] を基に DREC のためのデータセットとして ALFRED-fc を収集した．ALFRED-fc では，特定の物体を把持して特定の場所へ配置する “Pick & Place” に関するサブゴールにおける指示文，把持直前および配置直後のロボットの視覚画像を抽出した．ALFRED は物体操作を含む VLN における標準データセットであるが，ロボットが物体を運搬する際にカメラ画像中に把持している物体が空中に浮かんだ状態で写っている．これによりロボットの視界が遮蔽されてしまうため，物体操作の入力画像として不適切であった．そこで，本研究では物体を把持する直前および配置した直後のカメラ画像を収集した．

ALFRED-fc は，対象物体および配置目標に関するそれぞれ 1099 枚の画像，3452 文の英語で記述された指示文を含み，語彙サイズは 646，全単語数は 29113，平均文長は 8.4 である．全 5748 サンプルのうち，4420 サンプルを訓練集合に，642 サンプルを検証集合に，686 サンプルをテスト集合にそれぞれ使用した．なお，対象物体候補および配置目標候補に関するそれぞれの画像と指示文の組を 1 サンプルと定義する．訓練集合および検証集合は ALFRED における訓練集合から，テスト集合は valid seen および valid unseen から作成した．

物体配置直前のカメラ画像には把持中の物体が視界中央に写り込んでしまうため，物体配置直後の画像に対して配置した対象物体の領域を 0 埋めするマスク処理を行った．また，ALFRED には各画像における対象物体および配置目標に関する領域の Ground Truth (GT) が含まれるが，その他の対象物体候補および配置目標候補に関する領域はアノテーションされていない．そこで，Faster R-CNN [11] による物体検出を行うことで複数の領域を獲得し，GT との Intersection over Union (IoU) が 0.7 以上のものを正例，0.3 以下のものを負例を作成するために用いた．負例については作成方法が三通り存在する．第一に，対象物体候補に関して IoU が 0.3 以下のものを選ぶ方法，第二に，指示文を無作為に選んだ別サンプルのものに差し替える方法，第三に，これら両方を行う方法である．これは，配置目標が写る画像においては GT 以外に明確な配置目標候補が存在しない場合があり，タスクの難易度が下がることを防ぐためである．なお，負例の集合から正例と同じ数のサンプルを無作為に抽出することで，最終的な正例と負例のサンプル数を均一にした．

5.2 パラメータ設定

transformer [12] において，層数を $L = 2$，隠れ層の 1 層目の query，key，value の次元数を $H^{(1)} = (N+l+1) \times 768$，attention head 数を $A^{(1)} = 12$ と設定した．

表 1　ALFRED-fc における定量的結果

Method	Accuracy [%]
(i) Baseline	79.4 ± 2.76
(ii) Ours (w/o multi-task learning)	76.9 ± 2.91
(iii) Ours (w/o zero fill)	80.4 ± 5.31
(iv) Ours	$\mathbf{83.1 \pm 2.00}$

ただし，N は画像中の物体または配置先の数，l は指示文中の単語数を示す．最適化には AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) を使用し，学習率を 8×10^{-5}，ステップ数を 20000，バッチサイズを 8 とした．ここで，1 ステップは 1 つのバッチの処理を示す．また，Dropout の確率を 0.1，マルチタスク学習の各重み係数を $\lambda_{\text{targ}} = 1.0$, $\lambda_{\text{dest}} = 1.0$ とした．

提案手法の総パラメータ数は約 3277 万である．学習にはメモリ 24GB 搭載の GeForce RTX 3090 および Intel Core i9-10900KF を使用した．学習には約 20 分，推論には約 0.004 秒/sample を要した．学習中は，合計 20000 ステップのうち 2000 ステップごとに検証集合およびテスト集合による評価を行い，検証集合においてもっとも高い精度を記録したときのテスト集合における精度を，最終的な精度とした．

6. 実験結果

6.1 定量的結果

表 1 に，各手法の ALFRED-fc における精度を示す．実験は 5 回行い，精度はその平均値および標準偏差を示す．ベースライン手法は，MLU-FI タスクにおいて計算コストを削減しつつ高い精度を達成した Funnel UNITER [3] に対して，単純に配置目標候補を入力に追加することで DREC へ拡張したモデルとする．この手法では，対象物体候補および配置目標候補がともに対象物体および配置目標に一致するかを同時に推論する．

本実験で用いたデータセットは正例と負例のサンプル数が均等で偏りがないため，このような場合に標準的な精度を評価指標として採用した．ただし，提案手法では対象物体候補および配置目標候補について個別に推論を行う点でベースライン手法と異なるため，正解ラベル y を $y = y_{\text{targ}} \cap y_{\text{dest}}$，予測ラベル $\hat{y}$ を $\hat{y} = \hat{y}_{\text{targ}} \cap \hat{y}_{\text{dest}}$ と定義することで統一的な指標での比較を行った．ここで，y_{targ} は対象物体候補が対象物体であるかの真偽値，y_{dest} は配置目標候補が配置目標であるかの真偽値，$\hat{y}_{\text{targ}}$，$\hat{y}_{\text{dest}}$ はそれぞれ提案手法の target mode および destination mode における予測ラベルを表す．

表 1 より，ベースライン手法 (i) は精度が 79.4%であるのに対し，提案手法 (iv) は 83.1%となり 3.7 ポイント上回った．

6.2 Ablation Study

Ablation study として，以下の 2 条件を定めた．

(ii) w/o multi-task learning: マルチタスク学習の寄与を調べるため，一つのモデルで対象物体候補および配置目標候補に関するシングルタスクの学習を同時に行う．

(iii) w/o zero fill: Switcher における 0 埋めによる条件付けの寄与を調べるため，各モードで $\boldsymbol{x}_{\text{targ}}$ と $\boldsymbol{x}_{\text{dest}}$，$\boldsymbol{x}_{\text{targloc}}$ と $\boldsymbol{x}_{\text{destloc}}$ を同一の値にする．

表 1 に示すように，条件 (ii)，(iii) のモデルは提案手法を精度でそれぞれ 6.2，2.7 ポイント下回った．よって，マルチタスク学習と Switcher における 0 埋めによる条件付けのどちらも性能向上に寄与しており，特にマルチタスク学習の導入が有益であったことが分かる．

(a) 対象物体/対象物体候補　(b) 配置目標/配置目標候補
TP，指示文：“To place the soap on the rack.”

(c) 対象物体/対象物体候補　(d) 配置目標/配置目標候補
FP，指示文：“Put a towel in the bath tub.”

図 3　ALFRED-fc における定性的結果

6.3　定性的結果

図 3 に定性的結果を示す．赤色，橙色の矩形領域がそれぞれ対象物体および配置目標の GT，青色が対象物体候補および配置目標候補を表す．

(a)，(b) は True Positive (TP) の例である．この例において，ベースライン手法は $p(\hat{y}) = 3.14 \times 10^{-7}$ と出力した．一方，提案手法は $p(\hat{y}_{\text{targ}}) = 1.00$ および $p(\hat{y}_{\text{dest}}) = 1.00$ と出力しており，対象物体候補および配置目標候補の両方について正確な予測を行っている．

(c)，(d) は False Positive (FP) の例である．この負例は，対象物体候補に関して GT との IoU が 0.3 以下の領域を選ぶ方法で作成されたものであるため，配置目標候補に関しては GT に合致している．この例において提案手法は $p(\hat{y}_{\text{targ}}) = 0.999$ および $p(\hat{y}_{\text{dest}}) = 1.00$ と出力し，鏡に映るタオルが対象物体であると誤って予測した．原因としては，対象物体候補が鏡に映ったものであると理解することの難しさに加え，物体検出が不完全であり対象物体候補領域と GT との重なりが生じていることが考えられる．なお，物体検出精度の向上は本研究の対象外である．

6.4　エラー分析

テスト集合において，提案手法における失敗例は合計 257 サンプル存在した．このうち，False Positive が 24 サンプル，False Negative が 233 サンプルであった．

表 2 に，提案手法の評価における失敗例の分類結果を示す．精度を比較する際には，各モード 1 回ずつの推論により 1 サンプルの予測を行う．しかしこの方法では，$(y_{\text{targ}}, y_{\text{dest}}) = (0, 1)$ の組から生成した負例に対して $(\hat{y}_{\text{targ}}, \hat{y}_{\text{dest}}) = (1, 0)$ とモデルが予測した場合に正解と扱われてしまう．そこで，各モードについて個別に，それぞれ 50 の失敗例を人手で分析した．

失敗の要因は，SC，SO，SR，IVI，II，IL の 6 種類に大別される．SC は，提案手法が画像および言語の情報を適切に理解できなかったものを示す．SO は，対象物体と対象物体候補，または配置目標と配置目標候補が類似しているものを示す．SR は，候補領域が極端に小さいものを示す．IVI は，候補領域が物体または配置先を十分に包含できておらず視覚的な特徴が掴みづらいものを示す．II は，指示文の情報が不完全であるものを示す．IL は，データセットのラベル誤りを示す．

表 2　ALFRED-fc における失敗例の分類

Error ID	#Target Error	#Destination Error
SC	34	25
SO	8	7
SR	7	0
IVI	0	15
II	0	1
IL	1	2
Total	50	50

表 2 より，両モードに共通して SC が主要な失敗要因である．これに対して，画像および言語に関する理解力向上のため，汎用的な大規模データセットを用いて UNITER [10] で実施される Image-Text Matching などの事前学習を導入することが有効だと考えられる．

7.　おわりに

本研究では，物体操作タスクにおいて画像中の各物体および配置先の中から，指示文の対象物体および配置目標の両方を特定する DREC を扱った．

本研究の貢献を以下に示す．

- Funnel UNITER [3] に Switcher およびマルチタスク学習を導入することで，単一モデルで対象物体候補，配置目標候補のどちらも推論可能にした．
- 対象物体候補および配置目標候補に関する予測を個別に行うことで，少ない推論回数での対象物体および配置目標の探索を可能にした．
- ALFRED [1] を基にした DREC におけるデータセットである ALFRED-fc において，提案手法がベースライン手法を分類精度で上回った．

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. Shridhar, J. Thomason, D. Gordon, et al., “ALFRED: A benchmark for interpreting grounded instructions for everyday tasks,” CVPR, pp.10740–10749, 2020.
[2] S. Min, D. Chaplot, et al., “FILM: Following Instructions in Language with Modular Methods,” ICLR, 2022.
[3] 吉田悠，石川慎太朗，杉浦孔明，“生活支援ロボットによる物体操作タスクにおける Funnel UNITER に基づく指示文理解,” JSAI2022，p.2O4GS7，2022.
[4] Z. Dai, G. Lai, Y. Yang, and Q. Le, “Funnel-transformer: Filtering out sequential redundancy for efficient language processing,” NeurIPS, vol.33, pp.4271–4282, 2020.
[5] A. Mogadala, M. Kalimuthu, et al., “Trends in integration of vision and language research: A survey of tasks, datasets, and methods,” JAIR, vol.71, pp.1183–1317, 2021.
[6] A. Magassouba, K. Sugiura, and H. Kawai, “A Multimodal Target-Source Classifier With Attention Branches to Understand Ambiguous Instructions for Fetching Daily Objects,” RA-L, vol.5, no.2, pp.532–539, 2020.
[7] A. Magassouba, K. Sugiura, et al., “Understanding Natural Language Instructions for Fetching Daily Objects Using GAN-Based Multimodal Target–Source Classification,” RA-L, vol.4, no.4, pp.3884–3891, 2019.
[8] H. Fukui, T. Hirakawa, T. Yamashita, et al., “Attention branch network: Learning of attention mechanism for visual explanation,” CVPR, pp.10705–10714, 2019.
[9] S. Ishikawa and K. Sugiura, “Target-dependent UNITER: A Transformer-Based Multimodal Language Comprehension Model for Domestic Service Robots,” RA-L, vol.6, no.4, pp.8401–8408, 2021.
[10] Y.-C. Chen, L. Li, et al., “UNITER: Universal image-text representation learning,” ECCV, pp.104–120, 2020.
[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” Trans. PAMI, vol.39, no.6, pp.1137–1149, 2016.
[12] A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” NeurIPS, vol.30, pp.5998–6008, 2017.