Learning Bidirectional Translation between Descriptions and Actions with Small Paired Data

Minori Toyoda*1, Kanata Suzuki*1,2, Yoshihiko Hayashi1, and Tetsuya Ogata1,3,4

Abstract—This study achieved bidirectional translation between descriptions and actions using small paired data from different modalities. The ability to mutually generate descriptions and actions is essential for robots to collaborate with humans in their daily lives, which generally requires a large dataset that maintains comprehensive pairs of both modality data. However, a paired dataset is expensive to construct and difficult to collect. To address this issue, this study proposes a two-stage training method for bidirectional translation. In the proposed method, we train recurrent autoencoders (RAEs) for descriptions and actions with a large amount of non-paired data. Then, we fine-tune the entire model to bind their intermediate representations using small paired data. Because the data used for pre-training do not require pairing, behavior-only data or a large language corpus can be used. We experimentally evaluated our method using a paired dataset consisting of motion-captured actions and descriptions. The results showed that our method performed well, even when the amount of paired data to train was small. The visualization of the intermediate representations of each RAE showed that similar actions were encoded in a clustered position and the corresponding feature vectors were well aligned.

I. INTRODUCTION

Robots that collaborate with humans are required to have the ability to generate actions from language instructions and explain their actions. If this is the case, we would not need to devise a complicated system to command a robot. Besides, a human user could easily understand the behavior of the robot. To achieve this ability, the robot needs to associate real-world objects with linguistic expressions, and a comprehensive paired dataset is generally required to acquire that relationship. In particular, the machine learning-based approaches [1][2] require the preparation of a large paired dataset of descriptions and actions. However, collecting and captioning behavioral data is costly, making it difficult to collect large amounts of paired data.

Several joint learning methods for language and motion have been proposed to address these issues. Using context-independent word embeddings derived from a large corpus, [2] and [3] dealt with unseen words that were not included in the training data. In other studies, stepwise learning methods were proposed to integrate multiple models from different modalities [4], [5]. These previous studies demonstrated that pre-trained models trained using a large dataset can be effectively exploited by being fine-tuned using task-oriented data. Motivated by these works, we pre-train a model in each modality (language and motion) and fine-tune the entire bidirectional translation model using small paired data.

The current study proposes a two-stage training method that does not require a comprehensive paired dataset for bidirectional translation between descriptions and actions. An overview of this study is presented in Fig. 1. First, we pre-train two recurrent autoencoders (RAEs) for descriptions and actions separately with non-paired data, and then fine-tune the entire model by binding their intermediate representations using small paired data. The data used for pre-training do not need to be paired, behavior data and a large language corpus can be independently used for training those RAEs. Our experiments used a paired dataset of motion capture and reference caption to perform mutual generation of descriptions and actions. We performed quantitative and qualitative evaluations using large paired datasets, and an analysis of model behavior.

II. RELATED WORK

The present work aims at the bidirectional translation of descriptions and actions without the need for comprehensive paired data. The efficient use of non-paired and/or small data would be vital in this regard.

A. Utilization of Pre-trained Model in Translation Tasks

Many studies have proposed methods for training translation/ transformation models with limited data resources. One solution is to use a pre-trained model in the source/target modality. In the area of natural language processing, several combinations of source/target modalities have been worked out by using pre-trained models [4][6]. The quality and
quantity of the cross-modal data have been central issues to tackle in these studies. Using a pre-trained model in the source/target side could partly address the issue, particularly the quantity issue. In fact, large pre-trained language models, such as BERT [7], have significantly contributed to the improvement in the cross-modal task performances.

Several studies have been conducted on bidirectional translation tasks. With regard to the bidirectional text-to/from-image generation task, there are some studies that apply pre-trained models trained with large-scale data [8]. During the conversion between symbolic actions and texts in a 3D simulator environment [5], pre-training on large non-paired data has shown an improved performance in zero-shot settings. For description-from-to-action translation, [2] realized appropriate action generation that accepts unseen words not included in the dataset. They successfully retrofitted the pre-trained word embeddings into multimodal representations incorporating the action modality.

Creating a pre-trained model with a single modality is relatively easy, particularly by applying self-supervised learning. The effective use of a pre-trained unimodal model could address the issue of lacking paired data in translation/transformation, provided a proper integration of the representations in a particular modality is achieved.

B. Integration of Multimodal Representations

When data resources are limited, the method of integrating the two modalities, that is, accurate connection of the pre-trained models, is important. Multimodal representations can be categorized into two types according to the way they are acquired: joint and coordinated representations [9].

A joint representation is obtained by mapping all modalities into the same representation space [5][10]. In object manipulation and bell-ringing tasks, [10] obtained multimodal representations by concatenating the extracted representations such as images and sound signals with joint angles and passing them through an autoencoder. In this study, the feature extractor needs to be trained from scratch, which makes it difficult to train with a small amount of paired data. This requirement of large paired data is a general issue of joint representation [9].

A coordinated representation is obtained by enforcing similarity constraints and is often used in translation tasks [9]. The present study also adopts this method, following other studies of bidirectional translation between descriptions and actions [1][2]. A bi-modal autoencoder model has been proposed for the acquisition of multimodal representations of vision and language [11]. Appropriate multimodal word representations were obtained using pre-trained word embeddings and visual features, and learning to bring the autoencoder’s intermediate representations of word representations and visual features closer together. This constraint allowed us to train the entire model end-to-end at a lower computational cost than that of the joint representation.

To achieve efficient bidirectional translation between descriptions and actions with small amounts of data, we pre-trained the language and action models separately and then fine-tuned them by imposing a learning constraint that binds the intermediate representations of both modalities.

III. Method

A. Model Overview

In this study, we use a retrofitted paired recurrent autoencoder (rPRAE [2]) to perform bidirectional translation between descriptions and actions. An overview of the model is presented in Fig. 2. rPRAE comprises two RAEs for action and description, and a retrofit layer, which is a nonlinear layer that transforms word embeddings. Each RAE has a sequence-to-sequence model structure comprising encoder and decoder parts, and is trained to output the same data as the input time-series data. The encoder extracts low-dimensional feature vectors of the input data. In the rPRAE, the extracted feature vectors are input to the decoder of the other RAE to perform a bidirectional translation. In this paper, we refer to these feature vectors as intermediate representations.

In the action RAE, the joint angles and position/rotation of the body are used as the input and output of the model. Although the dataset used in this work excludes environmental information such as vision, representations that reflect the information necessary for dynamic description/action generation can be obtained by concatenating it with motion signals and inputting it to the action RAE as demonstrated in [2], provided that the experimental setting is simple.
The loss function $L_{\text{act}}$ of the action RAE is defined based on the squared error, as follows:

$$L_{\text{act}} = \sum_{i=1}^{T_a-1} \left\| \hat{z}_{t+1} - \tilde{z}_{t+1} \right\|^2_2,$$

where $T_a$ denotes the length of the input motion sequence, $\hat{z}$ denotes the predicted motion signal, and $\tilde{z}$ denotes the teaching signal.

In contrast, in the description RAE, the pre-trained word embedding is transformed by the nonlinear layer and input to the encoder. The decoder outputs the probability of occurrence of words in the vocabulary. The loss function of the description RAE, $L_{\text{dsc}}$, uses cross entropy and is defined as follows:

$$L_{\text{dsc}} = -\sum_{t=1}^{T_d-1} \sum_{w} x_{t+1}(w) \log y_t(w),$$

where $T_d$ is the length of the input description sequence, $W$ is the vocabulary size, $x$ is the word embedding input to the model, and $y$ is the output of the model.

In addition to the reconstruction errors of RAES, the rPRAE learns to approximate the intermediate representations output by the encoders. By binding the intermediate representations of both RAES, bidirectional translation can be performed by inputting the feature vectors extracted by one RAE encoder into the decoder of the other RAE. The loss function $L_{\text{bnd}}$ for the intermediate representation is defined as follows:

$$L_{\text{bnd}} = \sum_{i=1}^{B} \psi(z_{i}^{\text{act}}, z_{i}^{\text{dsc}}) + \sum_{i=1}^{B} \sum_{j \neq i} \max \{0, \Delta + \psi(z_{i}^{\text{act}}, z_{j}^{\text{dsc}}) - \psi(z_{i}^{\text{act}}, z_{j}^{\text{dsc}}) \},$$

where $B$ is the batch size, $\{z_{i}^{\text{dsc}} | 1 \leq i \leq B \}$ represents the intermediate representation of the description RAE, and $\{z_{i}^{\text{act}} | 1 \leq i \leq B \}$ is the intermediate representation of the action RAE. In this study, $\psi$ is the Euclidean distance between the feature vectors, but other losses can be used as well. $\Delta$ denotes the margin. The first term brings the corresponding intermediate representations closer, and the second term moves the non-corresponding ones further. This enables integrated learning to map intermediate representations of descriptions and actions in the feature space.

For the description RAE to extract intermediate representations similar to that of the action RAE, it is desirable that the input of the description RAE encoder also contains motion information. We also train the retrofit layer to output word vectors that include information on the action. Because of the alternating parameter updates of the RAE and retrofit layer described in the following subsection, the output of the retrofit layer has similar vector representations of words for each action meaning.

---

Algorithm 1: Training procedures for the rPRAE

**Require:** $X_{\text{dsc}}^{\text{NP}}, X_{\text{act}}^{\text{NP}}$: non-paired dataset

**Require:** $N_1, N_2, n_{\text{ch}}$: hyperparameters

1: randomly initialize learnable parameters: $\theta_{\text{rae}}, \theta_{\text{ret}}$
2: for $i = 1$ to $N_1$ do
   3: Sample non-paired data $x_{\text{dsc}}^{\text{NP}}, x_{\text{act}}^{\text{NP}}$ from $X_{\text{dsc}}^{\text{NP}}, X_{\text{act}}^{\text{NP}}$
   4: Calculate $L_{\text{dsc}}, L_{\text{act}}$ by forward-path
   5: Compute total loss: $L_{S1} \leftarrow L_{\text{dsc}} + L_{\text{act}}$
   6: if $int((i/n_{\text{ch}}))%2 = 1$ then
      7: Update $\theta_{\text{rae}}$ by gradient method w.r.t. $L_{S1}$
   8: else
      9: Update $\theta_{\text{ret}}$ by gradient method w.r.t. $L_{S1}$
10: end if
11: end for
12: make small paired dataset $X_{\text{P}}^{\text{dsc}}, X_{\text{P}}^{\text{act}}$ from $X_{\text{dsc}}^{\text{NP}}, X_{\text{act}}^{\text{NP}}$
13: for $i = 1$ to $N_2$ do
   14: Sample paired batch $\{x_{\text{dsc}}^{\text{P}}, x_{\text{act}}^{\text{P}}\}$ from $X_{\text{dsc}}^{\text{P}}, X_{\text{act}}^{\text{P}}$
   15: Calculate $L_{\text{dsc}}, L_{\text{act}}, L_{\text{bnd}}$ by forward-path
   16: Compute total loss: $L_{S2} \leftarrow L_{\text{dsc}} + L_{\text{act}} + L_{\text{bnd}}$
   17: if $int((i + N_1)/n_{\text{ch}}))%2 = 1$ then
      18: Update $\theta_{\text{rae}}$ by gradient method w.r.t. $L_{S2}$
   19: else
      20: Update $\theta_{\text{ret}}$ by gradient method w.r.t. $L_{S2}$
21: end if
22: end for

---

B. Two-Stage Training with Small Paired Data

This study proposes a two-stage training method for rPRAE that does not require a large amount of paired data. As shown in Algorithm 1, the proposed method comprises two training steps.

**Training stage 1:** First, we pre-train the RAES using only the reconstruction errors shown in equations (1) and (2). In this training stage, only non-paired data $X_{\text{dsc}}^{\text{NP}}, X_{\text{act}}^{\text{NP}}$ are required (line 3 in Algorithm 1). The loss function in training stage 1 $L_{S1}$ is defined as follows:

$$L_{S1} = L_{\text{dsc}} + L_{\text{act}},$$

Using independent loss functions in the RAES, each model is trained to extract action- and description-specific representations. In addition, it is possible to use non-paired datasets for training, which allows us to leverage uncaptioned actions and a large language corpus.

**Training stage 2:** After pre-training, we fine-tune the entire model using a small amount of paired data. Here, small paired data $X_{\text{dsc}}^{\text{P}}, X_{\text{act}}^{\text{P}}$ are obtained by captioning a portion of the non-paired data $X_{\text{dsc}}^{\text{NP}}, X_{\text{act}}^{\text{NP}}$ (line 12 in Algorithm 1). In addition to the reconstruction errors, the learning constraints in Equation (3) are applied to bring the intermediate representations of the RAES closer together. The loss function in training stage 2 $L_{S2}$ is defined as follows:

$$L_{S2} = L_{\text{dsc}} + L_{\text{act}} + L_{\text{bnd}},$$

This allows the RAES to retain their pre-trained representation while mapping each representation of the modality.
**IV. EXPERIMENTS**

**A. Dataset**

In this study, we performed experiments on the mutual generation of descriptions and actions with a large paired dataset: the KIT Motion-Language Dataset [13], which comprises motion-captured actions and reference captions. Examples of an action and descriptions are presented in Fig. 3. The KIT dataset contains 3911 motion sequences, each represented by an ordered sequence of 50-dimensional vectors: 44 joint angles of the whole body, three values of the body’s root position, and three values of its rotation. In the experiments, following the previous studies [1][14], each motion of 100–3000 steps was downsampled to $1/10^4$h of the sampling frequency. All the motion data values were normalized to the empirically determined range [-0.9, 0.9]. The KIT dataset also contains 6353 caption sentences with a vocabulary size of $W = 1646$. Each sentence consisted of 5–44 words, including the BOS and EOS, at the beginning and end. For each word, we extracted a 300-dimensional vector representation using a pre-trained Word2Vec [15]. Further, 80% of the available action sequences were used for training, 10% for validation, and the remaining 10% for evaluation (Table I). Because there were multiple captions for a single action, paired data were created for the number of captions. Therefore, the total number of pairs of data for training was 4368.

**B. Training Setup**

The configuration of each model is presented in Table II. LSTMs are used for the encoder and decoder parts of each RAE, and only the encoder of the description RAE uses a bidirectional LSTM [16]. Each LSTM consisted of one layer with 500 neurons, and only the decoder of the action RAE consisted of three LSTM layers. The retrofit layer consisted of three fully-connected (FC) layers with 400 neurons, and the input and output had 300 dimensions. The intermediate representation of each RAE has 500 dimensions. We used tanh as the activation function. The batch size $B$ was set to 80, and Adam [17] was used as the optimizer. The number of training epochs was set to 200 for training stage 1 and 400 for training stage 2. We trained 4368 sequences (described in Section IV-A); thus, when using the comprehensive training setup where all parameters are updated alternately for every certain number of $n_{ch}$ (lines 6–10 and 17–21 in Algorithm 1). This method refers to “chain-thaw” [12], which fine-tunes the parameters of the model one layer at a time to prevent overtraining of the retrofit layer due to an increase in training parameters. By training the layers separately, we obtain the word representations without overfitting to the small paired data while moderately limiting the expressive ability of each model. Additionally, the retrofit layer can map the word embedding to a similar feature space for each action [2].

Using the aforementioned two-stage training process, we obtain word embeddings that match the actions and update the parameters so that the feature space of each pre-trained modality is equalized with the small paired data. The proposed method has a simple structure; however, it can be applied to a variety of data types. The small paired data should be obtained in a similar setting to the non-paired data used in training stage 1 because training stage 2 requires binding the independent intermediate representations. This issue should be investigated further in future studies.

### Table I

| Number of Action | Number of Caption |
|------------------|-------------------|
| Train            | 2716              |
| Validation       | 272               |
| Test             | 273               |

### Table II

| Model          | Layer and Dimension                                                                 |
|----------------|--------------------------------------------------------------------------------------|
| action-RAE$^*$ | Input@50 - LSTM@500 - $z^{act}$@500 - LSTM@500 - Output@50                        |
| description-RAE$^*$ | Input@300 - BiLSTM@500 - $z^{dsc}$@500 - LSTM@500 - Output@1331                 |
| retrofit layer$^*$ | Input@300 - FC@400 - Output@300                                                   |

*) All layers have tanh activation, except for the transformation part of the intermediate representation $z^{act}$ and $z^{dsc}$. 

---

2We used only the paired data whose captions were formed with words equipped with the pre-trained embeddings. This selection of data resulted in a subset of 2721 actions and 5538 caption sentences. The vocabulary size is $W = 1331$; the max sentence length in the number of words is 38.
data in stage 1 and 500 paired items in stage 2, one epoch contains 55 and 7 iterations, respectively. Therefore, the total number of iterations for stage 1 is $N_1 = 11000$ and that for stage 2 is $N_2 = 2500$. In both training stages 1 and 2, the parameters of the RAEs and retrofit layer were updated by alternating every $n_{ch} = 100$ updates. These parameters were determined after considering several training settings, and there was no particular sensitivity problem if the training epochs $N_1$ and $N_2$ were large enough. We compared the performance of rPRAE without training stage 1 as the conventional method (vanilla rPRAE) with that of the proposed method, which performs both training stages.

C. Evaluation Metrics

An overview of the evaluation metrics used in the experiments is presented in Fig. 4. We varied the number of paired data of descriptions and actions, and verified the effectiveness of the proposed method from the results of two tasks: translation from action to description (Experiment 1) and translation from description to action (Experiment 2).

Experiment 1: We quantitatively evaluated the performance of the model for description generation using (1) multiple evaluation metrics for machine translation. In our experiments, we used BLEU [18], BERTScore [19], and cosine similarity of the sentence vectors obtained from SimCSE [20]. BLEU is a measure that calculates the agreement between the reference sentence and included words without considering context information. BERTScore uses a vector representation of tokens, including contextual information acquired from the language model BERT [7]; it has been reported to have a higher correlation with human ratings than BLEU. SimCSE is a model that uses BERT to embed the meaning of sentences in a more compatible way with human senses. For all of these scores, the higher the number, the closer the sentence is generated to the reference sentence.

Experiment 2: In the translation from description to action, quantitative evaluation is difficult because there is no fixed scale to evaluate each motion. Therefore, in this study, we evaluated the following points by referring to the conventional methods for bidirectional translation: (2-1) the back-translated sentences (description→action→description) using BLEU, BERTScore, and SimCSE as in Experiment 1, and (2-2) relative performance [14], which is the recovery rate of the generated action. Relative performance is the percentage of translation quality metrics that are retained after translating the description into action and then transforming this generated action back into a description using the model to be compared with. In our experiment, we used a reference model trained with all the paired data for comparison.

V. Results and Discussion

A. Experiment 1: Translation from Action to Description

First, we evaluated the proposed method in translation from action to description. The translation quality metrics of the generated descriptions are shown in Table III. The first and second columns show the number of paired and non-paired data used for training, respectively. The top four rows show the conventional method (vanilla rPRAE) trained with paired data only, and the bottom three rows show the proposed method pre-trained on non-paired data and fine-tuned on paired data. Each result shows the average of three training trials using random seeds. In the vanilla rPRAE, all non-paired data numbers were zero because pre-training was not performed.
Fig. 6. Generated samples of actions by inputting each description: (I) and (II). The proposed model was pre-trained with 4368 non-paired data and fine-tuned with 500 paired data, same as Fig. 5. Vanilla rPRAE was also trained with 500 paired data. (a) shows the images of the generated actions, and (b) and (c) show their joint angles and rotations, respectively. Positions are not shown due to space limitations, but the results are similar to (b) and (c). The solid lines indicate the generated values. The broken lines indicate the correct trajectories. Among the 44 joint angles, eight of them are shown in (b).

| TABLE V | COMPARISON OF CORES OF BACK-TRANSLATED DESCRIPTIONS BY AMOUNT OF PAIRED DATA |
|---------|--------------------------------------------------------------------------------------------------|
| Paired  | Non-Paired | BLEU | P_BERT | R_BERT | F_BERT | SimCSE |
|---------|-------------|------|--------|--------|--------|--------|
| 4368    | 0           | .259 | .905   | .901   | .903   | .648   |
| 500     | 0           | .228 | .900   | .898   | .899   | .600   |
| 250     | 0           | .227 | .899   | .897   | .898   | .588   |
| 100     | 0           | .217 | .899   | .896   | .897   | .577   |
| 4368    | 500         | .242 | .903   | .899   | .901   | .618   |
| 250     | 4368        | .235 | .902   | .898   | .900   | .604   |
| 100     | 4368        | .232 | .903   | .897   | .900   | .595   |

When comparing the methods with the same amount of paired data, it was confirmed that the proposed method produced sentences that were closer to the correct sentences than the vanilla rPRAE as the amount of paired data became smaller. This tendency was observed regardless of the type of score, and when the amount of paired data was less than 250 (approximately 5.7% of the total data), the proposed method outperformed the vanilla rPRAE in all cases. However, when the amount of paired data exceeded 500 (approximately 11.4% of the total data), the vanilla rPRAE rarely outperformed the proposed method. This is presumably because 500 paired data are almost sufficient for the KIT dataset, and the impact of pre-training using non-paired data became smaller.

To verify the effect of non-paired data on the proposed method, we investigated the scores shown in Table [V]. Except for the number of non-paired data, the training settings remained unchanged. The overall trend was that the smaller the non-paired data, the lower the score. When the amount of non-paired data was 750 (the fourth row of Table [III]), it was confirmed that errors at the word level occurred frequently, although grammatically appropriate sentences were generated. Therefore, the translation quality metrics can be improved by collecting non-paired data, which indicates the effectiveness of the proposed method using an independent pre-trained model.

In addition to the quantitative evaluation using translation quality metrics, we also conducted qualitative evaluations by...
visually checking the generated descriptions. We confirmed appropriate description generation for actions in which the vanilla rPRAE failed to translate correctly (Figure 5). This is because of the improvement of the expressive ability in the model by pre-training using non-paired data, as described earlier. However, there were some cases in which the proposed method failed to generate the data while the vanilla rPRAE succeeded. It is often possible to confirm that the vanilla rPRAE generates appropriate actions in cases where similar motions are included in the training data. From these results, it can be inferred that, although the vanilla rPRAE was originally powerful in verbalizing behaviors included in the training data, the proposed learning method improved the generalization ability as a trade-off.

B. Experiment 2: Translation from Description to Action

Next, we evaluated our method from the viewpoint of translation from description to action. Table VII shows the results of the back-translation evaluation of the generation actions. Table VII shows their relative performance to the reference model (first row in Table III) as illustrated in Fig. 4.

The overall score tends to decline as the amount of non-paired data increases, suggesting that collecting non-paired data can improve the quality of the generated sentence. In addition, the score of the proposed method was much higher than that of Experiment 1, indicating that the proposed method is particularly suitable for translation from description to action. This is because the representations of the actions acquired by pre-training were more diverse than those of the descriptions.

As in Experiment 1, we conducted a qualitative evaluation by visually checking the generated samples. In most settings, the proposed method generated actions comparable to those of the reference model, whereas the vanilla rPRAE generated many cases of indistinguishable actions. Figure 6 showed that the proposed method repeatedly generated proper actions, although the start timings were different from the correct ones. In summary, the proposed method can appropriately bind the intermediate representations of the description and action RAEs by fine-tuning with small paired data. Moreover, their bidirectional transformation was realized. This study confirms the effectiveness of the proposed two-stage training concept. However, owing to the simplicity of the proposed method, there is room for improvement, such as the use of non-paired data from different settings and the introduction of metric learning for clustering using positive/negative information [21][22]. The latter is considered an effective method based on the visualization results of the intermediate representations described in the next subsection.

C. Comparison of Shared Representations

Finally, to analyze the model behaviors, the intermediate representations of both the encoded descriptions and actions were visualized using t-SNE. Figure 7 (a) shows the representation $z^{\text{act}}$ by the action RAE, and Fig 7 (b) shows the representation $z^{\text{dsc}}$ by the description RAE. The visualized model was trained with 500 paired data and 4368 non-paired data. The plots in the figure show the encoded actions in the test data with the labels “walk,” “wipe,” “play,” and “dance,” which are frequently occurring verbs. In addition, samples with “walk” contained supplementary information about the direction of movement, such as “forward,” “turn,” and “circle.”

Although the boundaries were mixed, the samples with the same label were encoded in a similar position in the intermediate representation of each RAE. In particular, movements related to “walk” and those related to the upper body (“wipe,” “play,” “dance”) were clearly separated. Additionally, within the actions with the “walk” label, only apparently distinctly different actions (“circle”) were appropriately separated. Therefore, it was confirmed that the intermediate representations of each RAE were bound, even when fine-tuning was
performed with small paired data. Some of the generated samples have complex structures in the feature space. In many cases of “walk”+“forward” and “walk”+“turn,” they encompassed multiple motions, which were widely scattered in the feature space. These two actions were considered to overlap because the beginnings of their motions were similar. In addition, because the description RAE reflected factors other than the verb labels employed in this analysis, the samples were not clearly grouped compared to the action RAE. The results were also significantly affected by orthographical variants in the captions, particularly for the “dance” label samples. Additionally, the words “right” and “left” were used in “walk”+“turn,” “wipe,” and “play” to indicate the direction of movement or the moving hand, which may have affected the structuring in the feature space. This may also be due to the vector representation of Word2Vec, which does not consider contextual information. We believe that utilizing the different properties of these intermediate representations in the generation phase is the next step toward improving the model.

VI. CONCLUSION

This study proposed a learning method for bidirectional translation between actions and descriptions using small paired data. In the proposed method, RAEs for actions and descriptions were trained on large-scale non-paired data, and then the entire model was fine-tuned on small-scale paired data. As a result of the experiments on the mutual generation of actions and descriptions, it was confirmed that the proposed method could realize appropriate bidirectional transformation even if the paired data connecting them are small. The visualized intermediate representations of each RAE showed that similar actions were encoded in a clustered position and the corresponding feature vectors were well aligned. We could use any encoder–decoder model for each modality because their structures are not constrained.

The use of context-independent word embeddings, however, had the limitation that it may not be able to properly handle polysemous expressions. We plan to improve the performance of our method by incorporating more powerful context-dependent language models, such as Transformer [23]. Furthermore, because the interaction with various objects and controlling the timing to generate speech and actions are essential to realize a collaborative robot, we would like to collect data and address these issues using the robot AIREC.

ACKNOWLEDGMENT

This work was supported by JST [Moonshot R&D][Grant Number JPMJMS2031].

REFERENCES

[1] T. Yamada, H. Matsunaga, and T. Ogata, “Paired recurrent autoencoders for bidirectional translation between robot actions and linguistic descriptions,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3441–3448, 2018.

[2] M. Toyoda, K. Suzuki, H. Mori, Y. Hayashi, and T. Ogata, “Embodysing pre-trained word embeddings through robot actions,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 4225–4232, 2021.

[3] D. Matthews, S. Kriigman, C. Cappelle, and J. Bongard, “Word2vec to behavior: morphology facilitates the grounding of language in machines,” in Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019, pp. 4153–4159.

[4] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained models for natural language processing: A survey,” Science China Technological Sciences, vol. 63, no. 10, pp. 1872–1897, Sep 2020.

[5] R. Zellers, A. Holtzman, M. Peters, R. Mottaghi, A. Kenilvati, A. Farhadi, and Y. Choi, “Piglet: Language grounding through neuro-symbolic interaction in a 3d world,” in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021, pp. 2040–2050.

[6] L. Ruan and Q. Jin, “Survey: Transformer based video-language pre-training,” AI Open, vol. 3, pp. 1–13, 2022.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre training of deep bidirectional transformers for language understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, 2019, pp. 4171–4186.

[8] Y. Huang, B. Liu, and Y. Lu, “Unifying multimodal transformer for bi-directional image and text generation,” in Proceedings of the 29th ACM International Conference on Multimedia, 2021.

[9] T. Baltrutsaitis, C. Ahuja, and L.-P. Morency, “Multimodal machine learning: A survey and taxonomy,” IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 2, pp. 423–443, 2018.

[10] K. Noda, H. Arie, Y. Suga, and T. Ogata, “Multimodal integration learning of robot behavior using deep neural networks,” IEEE Robotics and Autonomous Systems, vol. 62, no. 6, pp. 721–736, 2014.

[11] M. Hasegawa, T. Kobayashi, and Y. Hayashi, “Incorporating visual features into word embeddings: A bimodal autoencoder-based approach,” in Proceedings of the 12th International Conference on Computational Semantics, 2017.

[12] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,” in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1615–1625.

[13] M. Plappert, C. Mandery, and T. Asfour, “The KIT motion-language dataset,” Big Data, vol. 4, no. 4, pp. 236–252, 2016.

[14] ——, “Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks,” IEEE Robotics and Autonomous Systems, vol. 109, pp. 13–26, 2018.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” Advances in neural information processing systems, vol. 26, 2013.

[16] D. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[19] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” in Proceedings of the International Conference on Learning Representations, 2020.

[20] T. Gao, X. Yao, and D. Chen, “SimCSE: Simple contrastive learning of sentence embeddings,” in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 6894–6910.

[21] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, “Big self-supervised models are strong semi-supervised learners,” arXiv preprint arXiv:2008.10029, 2020.

[22] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” arXiv preprint arXiv:2003.04297, 2020.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, vol. 30, 2017.