Cross-modal Representation Learning for Understanding Manufacturing Procedure

Atsushi Hashimoto[1][0000−0002−0799−4269], Taichi Nishimura[2], Yoshitaka Ushiku[1][0000−0002−9014−1389], Hirotaka Kameko[2][0000−0001−9844−6198], and Shinsuke Mori[2]

[1] OMRON SINIC X Corp., Tokyo, Japan
atsushi.hashimoto@sinicx.com
https://www.omron.com/sinicx/en/
[2] Kyoto University, Kyoto, Japan

Abstract. Assembling, biochemical experiments, and cooking are representatives that create a new value from multiple materials through multiple processes. If a machine can computationally understand such manufacturing tasks, we will have various options of human-machine collaboration on those tasks, from video scene retrieval to robots that act for on behalf of humans. As one form of such understanding, this paper introduces a series of our studies that aim to associate visual observation of the processes and the procedural texts that instruct such processes. In those studies, captioning is the key task, where input is image sequence or video clips and our methods are still state-of-the-arts. Through the explanation of such techniques, we overview machine learning technologies that deal with the contextual information of manufacturing tasks.

Keywords: Procedural Text Generation · Image Captioning · Video Captioning · Understanding manufacturing activity.

1 Introduction

The versatile fitting performance of deep neural networks has established a new paradigm, cross-modal processing, typified by image captioning. It is a natural ability for humans to explain what we witnessed by language or, inversely, imagine a scene from a text description. Such vision-language abilities are generalized as a projection of physical world observation into a symbolic world and vice versa.

The decisive difference between cross-modal processing and traditional classification problem is the non-existence of a format for symbolic expression. A traditional framework must have a pre-defined output format, limiting possible output patterns unnecessarily. In contrast, a linguistic expression has no such limitation. In this sense, captioning, for example, is an ultimate task for a machine learning (ML) model to extract information from a visual observation under no external limitations.

We focus such captioning tasks on manufacturing, an activity to produce a valuable product from multiple materials through multiple processes. In manufacturing, the processes are described as procedural text. Humans can reproduce a product as long as the processes are described appropriately. Such an ability is vital for a system that displays instruction as work progresses by matching the situation and instruction. Similarly, it contributes to realizing a robot that obeys textual instruction and executes manufacturing; given instruction is compared with the current situation to identify the required robot's physical actions indicated by the textual instruction.

Considering procedural text generation from visual observation, the major difference from general captioning tasks is two-fold; it has a flow of material combination, and the materials change their state through processes. This paper introduces two techniques that model each of them: the state-of-the-arts to obtain cross-modal representation in manufacturing applications. Note that, due to the dataset availability, we evaluated those methods on the dataset of cooking activities.

2 Captioning on time series of visual observation

Since the success of image captioning by deep learning [29], its architecture of encoder (feature extractor) and decoder (text generator) has extended for other visual formats; visual story telling [8] and video paragraph captioning [31]. This section overviews manufacturing activity understanding from the viewpoints of these two problems.

2.1 Visual storytelling

Visual storytelling is a task to generate a text for each image in a sequence, firstly proposed in [8]. This task is the most simple extension of image captioning to a time series observation; it should avoid duplicated mentions or track entities to refer to them appropriately. In manufacturing, the model should also consider an additional temporal context; how the materials have changed their state between two consecutive images. To model such changes, Chandu et al. [3] projected the latent space into discrete states and tried to imitate the state transition of real procedural text on a finite state machine. Two different approaches were made by us [19, 21]. In [19], we have proposed a method that pre-trains the encoder by state-wise image-text retrieval. Training a model to discriminate state difference and use the retrieved sentences as references, we obtained more accurate description of each step than [3]. In [21], we have explicitly modeled the process of material mixing, which contributes to enhance the captioning performance independently with the other methods. We introduce the details of this method in this paper. Note that, in our method, we assume a given list of materials since it is often indistinguishable even for humans (e.g., white powders or clear liquid) without labels.

Since there are a number of how-to web contents with images, there are several datasets for this task in manufacturing, mainly in cooking [6, 32, 3].

2.2 Video paragraph description

Video paragraph description is a task to generate a text for each video segment [31], which makes a paragraph of a video. As visual storytelling, the primal challenge of this task is to reflect the context of the event sequence. Generally, a video clip does not focus on entities of interest as much as an image. In addition, it may include actions that an image can hardly represent. Hence, a spatio-temporal attention mechanism and action recognition should be considered with contextual information. When video paragraph description is firstly defined, segments are not given, and a model must identify segments of interest simultaneously with paragraph description [31, 35]. However, due to its hardness, the problem was later re-defined as a simpler task with given segment boundaries [24, 13]. In [20], we focused on state transition caused by actions observed in each video segment, with given segment boundaries and material lists. We introduce the details of this method in this paper, as well as [21]. In parallel with our studies, Shi et al. [25, 26] proposed a method that utilizes transcript of narrated videos for generating fine-grained procedural text. Transcripts are a powerful resource to solve this task, but it is not always available because adding a transcript to a video requires an effort. Instead, our studies assume a given list of materials as a minimum external resource.

Since creating how-to content with videos is not a lightweight task, there are fewer datasets than images. YouCook2 [34] is almost only the dataset for this task, but the HowTo100M dataset [17] is often used for pre-training an encoder. We have also published a dataset with bio-chemical tasks [22], but the number of videos is quite limited. Note that there is a cross-modal dataset, YouMakeup [30], which is organized with videos of face makeup and instructions. This is slightly different from manufacturing because it is a task to paint cosmetics for different face parts rather than combining materials.

3 Procedural text generation by modeling material merging process

This section introduces our method that models merging processes of manufacturing, which is implemented on the problem of visual storytelling.

3.1 Tree representation of the merging processes

Since manufacturing is a task to yield a valuable product from materials, it is essential to merge multiple materials. Among traditional graph representations of manufacturing process [9, 10, 18, 33], we adopted Simplified Ingredient Merging Map in Recipes (SIMMR) [9], a graph representation of the manufacturing process whose node and edge models only material-merging actions. Based on SIMMR and the Cookpad Image Dataset [6], we have prepared a cross-modal dataset, visual SIMMR (vSIMMR) (Fig. 1).

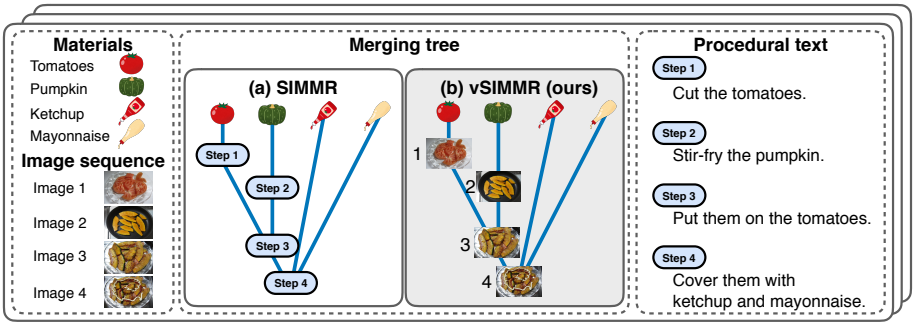


Fig. 1. The difference between SIMMR and vSIMMR (cited from [21]). Leaves of a tree graph refer to materials, and the other nodes refer to step-wise instructions in both SIMMR and vSIMMR. In addition, vSIMMR has an image corresponding to each step.

Each sample of this dataset consists of step-wise instructional texts, step-wise images, and a material list. Its annotation is given as a tree, whose leaves refer to materials in the list, and the other nodes refer to step-wise text and images.

3.2 Encoder-decoder model that models merging processes

Based on the vSIMMR dataset, we aim to develop an encoder-decoder model that yields procedural text from given images and a material list. The overview of the model is shown in Fig. 2. There are encoder and decoder modules that consider the merging process. The additional tree-reprediction module aims to allow semi-supervision of the tree structure. We assume a situation where we can access a large number of image-instruction pairs collected on the web (the Cookpad Image dataset) and only a few percent of data with tree-structure annotation (the vSIMMR dataset). The following parts explain how the structure models merging process of manufacturing and how semi-supervised learning is realized. Please refer to the original paper [21] for a mathematically rigorous explanation.

Text generator that considers the tree structure Utilizing this structure, we constructed a text generator (Process (iii) in Fig. 2) based on the child-sum tree LSTM [27]. The tree LSTM receives hidden states from multiple elements, children on a given tree structure. Thus, our decoder explicitly models the merging process of manufacturing.

Since the tree structure is not given at inference (other than leaf nodes of materials), we need to estimate the structure before calculating the decoder. We calculate attention-like weights for each material-image pair and image-image pair to estimate the edges, which we will explain in the next part. Regardless of the link weight calculation methods, we need to determine a tree structure by

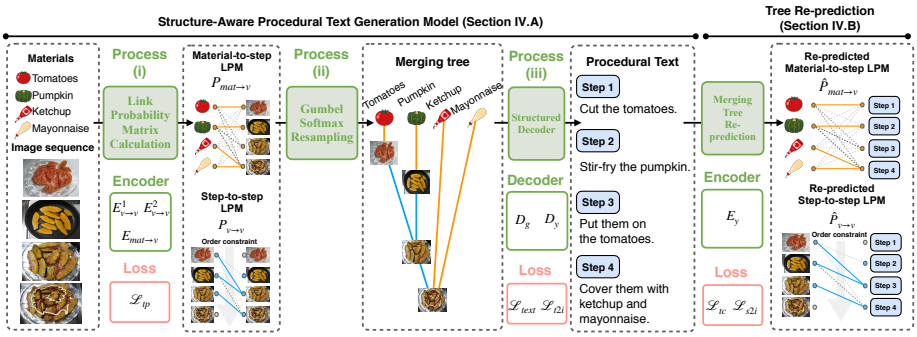


Fig. 2. The overview of proposed model (cited from [21]).

selecting edges based on the weights (Process (ii) in Fig. 2). Such deterministic operation usually breaks the chain rule of differentiation and disable end-to-end optimization, which is crucial for neural networks to fulfill their potential. Hence, at estimating the tree structure, we use the Gumbel-max trick [5].

Material and image encoders with tree structure prediction Each material in the list is first embedded into a latent feature by a word2vec encoding. Then, it is further embedded through bi-LSTM, where all materials are input to consider the context in the list. We refer to the embedded feature of materials as z_m.

Similarly, each image is embedded by a ResNet50 model [7] pre-trained with an image-step retrieval task on the Cookpad Image dataset. Then, they are further embedded by two different LSTMs to obtain key and query features, z_v^k and z_v^q.

Let $z_v^{\max}$ be a feature obtained from z_v^k and z_v^q with an element-wise max-pooling. Calculating similarity $S(z_v^{\max}, z_m)$ and $S(z_v^k, z_v^q)$, we obtain the material-step link weights and step-step link weights. z_m and z_q are re-used as the node feature of the tree at the decoder.

Tree-reprediction module for semi-supervised learning We assume semi-supervision for tree structure prediction since tree annotation is costly. This means that we can calculate the loss for tree prediction only a few percentages of training data, which is generally hard to cover diverse recipes. To assist the prediction, we use a principle that the tree structure implied from an image sequence (input) and the procedural text (output) must be identical. Based on this principle, we add a tree-reprediction module that predicts the same tree structure from generated texts. We can regard the decoder and this module as tree auto-encoder. Since the decoder is based on Tree-LSTM, a better tree prediction facilitates a better text generation. Similarly, better text generation ensures a better tree-reprediction. Hence, we add loss for samples without tree

structure annotation that evaluate inconsistency of re-predicted tree against the original estimation.

We add two βVAE-like modules that predict image features from the tree and step-wise sentences as additional modules for semi-supervision. For more details, please see the original paper [21].

3.3 Experiments

In this section, we introduce some of the key results of our method. Please refer to the original paper for the detailed conditions of experiments.

From the Cookpad Image dataset, we discarded recipes with steps without images, less than two steps, or less than two ingredients. After this selection, we got 200k recipes with complete images, more than one step, and one ingredient. We divided them into train/val/test sets with a ratio of 80%/10%/10%. We have annotated the tree structure for c.a. 1% of samples in each set.

Since our method is compatible with many existing methods for visual storytelling, we implemented our method on the following four methods.

Image2seq [8] is the method firstly proposed for the visual storytelling task. It involves the temporal context with BiLSTM after encoding each image independently.

GLAC Net [11] considers the context by global and local image feature vectors.

SSiD and **SSiL** are methods both proposed in [3]. It projects continuous embedded features into discrete state space, where a finite state machine (FSM) models state transition. Feed-forwarding state information to the text generation enriches temporal contextual information (SSiD). SSiL is an extension of SSiD, which has an additional loss to better imitate the state transition of ground truth.

RetAttn [19] is the state-of-the-art method of visual storytelling on manufacturing activities. It cross-modally retrieves the ten most similar sentences for each step and concatenates the average feature of retrieved texts to visual feature for text generation.

As an ablation study, we prepared the **full model** and **half model**, which are with and without the tree re-definition module.

Table 1 shows the key results of the proposed method. Here, BLEU [23] and ROUGE-L [15] are major metrics for machine translation and captioning. Distinct [14] is a metric to evaluate diversity of description. Ingredient/action are scores calculated on sequences obtained by extracting only ingredient/action from generated texts, which show coherency of generated recipes. Overall, the table shows the proposed method's versatile ability of performance enhancement.

Fig. 3 shows an example of generated procedural text and the tree structure predicted as a bi-product. In this example, our full model achieved an impressive description, "mix seasonings marked • in the ingredient list." "*bullet*" would be useful to re-predict tree structure from the text since it directly refers to the leaf nodes. This observation is supported by the predicted tree structure, where half model failed to predict the tree, but the full model succeeded.

Table 1. Word-overlap metrics for the five base models with half and full models. The scores in bold are the best for each base model. B=BLEU, RL=ROUGE-L, D=Distinct, I=Ingredient, and Ac=Action. * indicates statistically significant difference ($p < 0.001$) from the base models (original) through bootstrap sampling [12]. (cited from [21])

	B4	RL	D1	D2	I-B3	I-B4	I-RL	Ac-B3	Ac-B4	Ac-RL
Images2seq	5.1	18.4	38.3	54.7	0.5	0.1	9.7	3.8	2.0	18.4
+ Half	5.8*	20.6*	**51.1***	**75.0***	0.7*	0.2	12.7*	3.9	2.0	21.2*
+ Full	**6.3***	**21.7***	47.6*	71.0*	**0.9***	**0.3***	**13.8***	**4.3***	**2.1**	**22.9***
GLAC Net	5.9	21.4	46.6	69.0	0.9	0.3	13.2	4.3	2.1	22.8
+ Half	5.9	**21.8***	46.7	68.8	1.1*	**0.4**	15.6*	**4.4**	2.3	**23.1**
+ Full	**6.1**	21.3	**47.2***	**69.9***	**1.2***	**0.4**	**16.3***	**4.4**	2.3	22.5
SSiD	6.0	20.9	45.5	66.6	0.8	0.2	13.1	4.0	2.1	21.6
+ Half	6.2*	20.8	43.9	65.1	**1.3***	**0.4***	16.4*	4.4*	**2.2**	22.1*
+ Full	**6.4***	**21.6***	**48.3***	**71.0***	1.2*	**0.4***	**16.7***	**4.5***	**2.2**	**23.5***
SSiL	6.3	21.4	45.5	66.8	0.7	0.2	12.5	4.0	2.0	22.1
+ Half	5.4	21.4	46.7*	68.2*	1.2*	**0.4***	16.9*	3.8	2.0	22.2
+ Full	**6.4**	**21.9***	**47.3***	**70.9***	**1.4***	**0.4***	**17.0***	**4.5***	**2.3***	**23.0***
RetAttn	6.5	21.6	40.2	60.3	1.0	**0.3**	14.5	3.2	1.5	21.2
+ Half	6.5	21.8	52.4*	77.8*	**1.2**	**0.3**	**14.8**	**4.2***	2.0*	22.9*
+ Full	**7.1**	**22.1**	**52.7***	**78.6***	**1.2**	**0.3**	**14.8**	**4.2***	**2.0***	**23.1***

4 Procedural text generation by modeling state transition in continuous latent space

This section introduces our method that models changing state of materials through processes of manufacturing. Such state change was modeled in SSiD/SSiL [3]. However, its projection from a continuous latent space to the discrete state machine model must have unavoidable information loss. It is also difficult to identify the optimal number of states. When modeling state transition directly in a continuous space, we do not face these problems. To learn such continous state transision, we proposed a model that update latent feature in an action-driven manner. To observe actions, we implemented the model on the video paragraph description task.

4.1 Simulation of materials state transition

We model the material's state transition based on Neural Process Network (NPN) [2]. NPN is a model originally proposed for procedural text understanding, where input is procedural text and simulate the change of the states for each entity with the verbs in the sentence. We replace the entities with materials, which is given as the ingredient list. Similarly, we replace verbs in the text with actions observed in video segments, where we train an action recognition

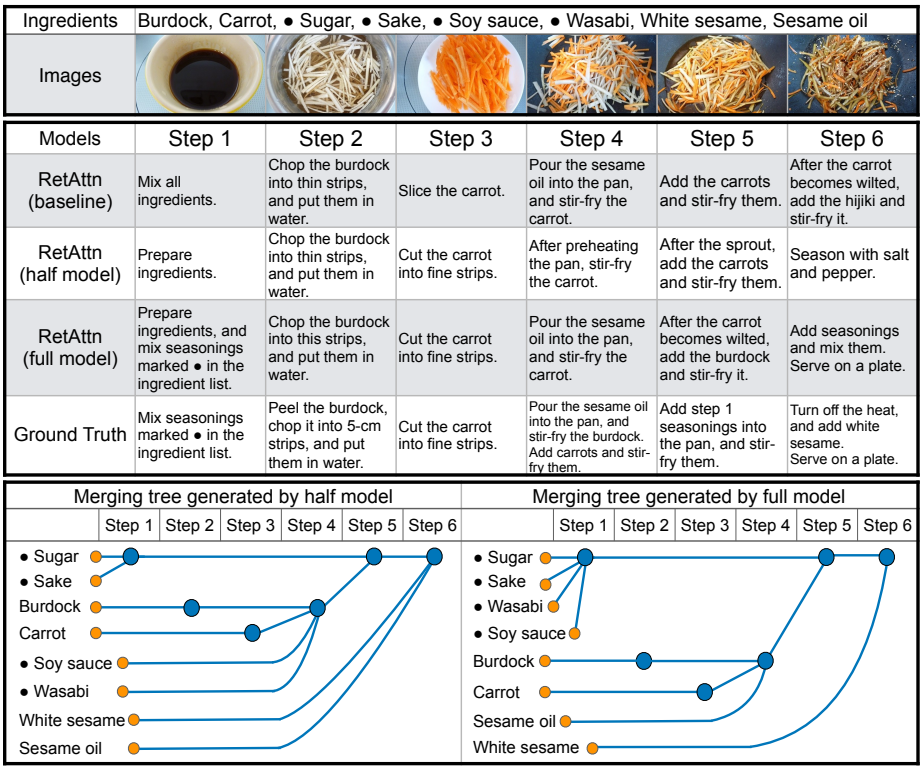


Fig. 3. Example of generated recipes and merging trees. Here, the baseline (original), half, and full models are compared with the ground truth. This sample has no ground truth of the merging tree. Note that the recipes are originally in Japanese and have been translated into English. (cited from [21])

model (and an involved material recognition model) with a distant supervision with procedural texts.

Fig. 4 shows the action-driven operation on latent features of materials, which we call visual simulator. The input of this simulator is embedded features of materials and the n-th clip (=video segment). When $n = 1$ (at the first step), the material features are obtained as the method in the previous section. When $n > 1$, material features are those updated at n-th step. To train action and material selectors, we use distant supervision. Namely, we extract material names and verbs from the text for each step and use them as ground truth. Note that this supervision is unreliable, especially at later steps, due to zero anaphora or missing reference.

Based on the principle of identical state transition between visual observation and procedural text, we re-simulate the state transition on the generated text and minimize the inconsistency between visual and textural simulations. Please refer to the original paper for the architecture of the entire model [20].

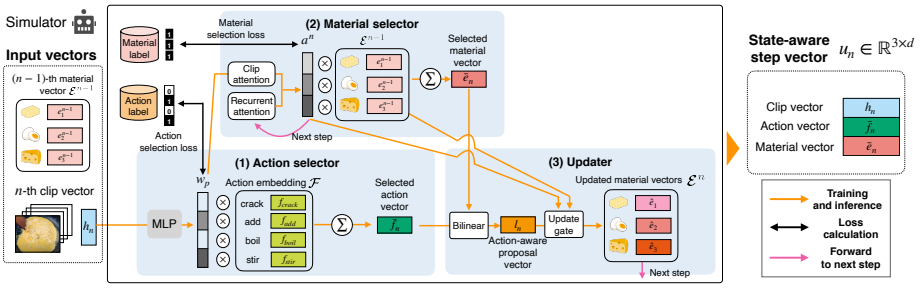


Fig. 4. An overview of the visual simulator. The simulator recurrently reasons the state transition of the materials at each step. Specifically, it predicts executed actions and involved materials in (1) the action and (2) material selectors. The selected material features are updated based on the selected action features. This update operation simulates the state transition in a continuous latent feature space (cited from [20]).

Table 2. Paragraph- and sentence-level word-overlap evaluation for the baseline and the proposed models with ablation studies. The scores in bold are the best among the comparative models. "I" indicates whether the model uses ingredient information or not. M=METEOR, C=CIDEr-D (cited from [20]).

Baseline	I	B1	B2	B3	B4	M	C	RL
Transformer-XL		39.0	22.0	12.1	6.7	15.2	22.7	30.9
+ Ingredients (Transformer-XL-I)	✓	37.7	22.5	13.4	8.2	15.4	35.4	34.2
MART		37.9	21.7	12.4	7.6	15.0	29.1	32.3
+ Ingredients (MART-I)	✓	42.3	26.2	16.1	9.9	17.6	48.2	36.2
Ours								
Video only (V)		43.2	24.5	14.0	8.1	16.6	32.4	31.9
V + Ingredients (VI)	✓	49.1	29.5	17.6	10.5	20.3	63.3	35.2
VI + Visual simulator (VIV)	✓	**49.4**	30.1	18.0	11.0	21.0	66.1	36.8
VIV + Textual re-simulator (VIVT)	✓	**49.4**	**30.9**	**18.3**	**11.3**	**21.1**	**67.1**	**37.1**

4.2 Experiments

We evaluated the above model on the YouCook2 dataset. Since the dataset does not have material lists, we prepared them by ourselves, which is accessible at https://github.com/misogil0116/svpc.

We compared our method with Transformer-XL [4] and MART [13], as the state-of-the-arts for general text generation tasks and the video paragraph description task, respectively. For a fair comparison, we prepared "+Ingredients" models for them (see the appendix of the original paper for more details). In addition to BLEU and ROUGE-L in Table 1, we show the scores of METEOR [1] and CIDEr-D [28]. Table 2 shows a clear superiority of our method against the baselines.

Ingredients	flour, eggs, baking soda, salt, pepper, water, shrimp, batter, breadcrumbs, oil		
	step 1	step 2	step 3
Clip sequence			
MART + Ingredients (MART-I)	add **flour salt** and **pepper** to a bowl and mix (✗ eggs, baking soda)	add milk egg and milk to the bowl and mix (✗ water)	coat the dough in the **batter** (✗ shrimp, breadcrumbs)
V + Ingredients (VI)	mix **flour salt pepper** and breadcrumbs (✗ baking soda, eggs)	mix flour salt pepper and breadcrumbs with the flour (✗ water)	coat the **shrimp** with the flour mixture (✗ batter, breadcrumbs)
VI + Visual simulator (VIV)	mix **flour eggs** and **salt** together (✗ baking soda, pepper)	add salt pepper to the eggs and mix (✗ water)	coat the **shrimp** in the **batter** (✗ breadcrumbs)
+ VIV + Textual re-simulator (VIVT)	mix **flour eggs baking soda salt** and **pepper** and **salt**	add **water** eggs breadcrumbs to a bowl of **water** and mix	coat the **shrimp** in the **batter** (✗ breadcrumbs)
Ground truth	add **flour eggs baking soda salt** and **pepper** to the bowl and stir	add cold **water** to the bowl and stir	cover the **shrimp** in the **batter** and **breadcrumbs**

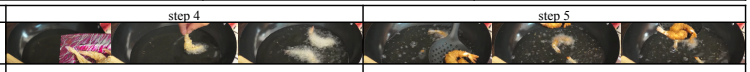
	step 4	step 5
Clip sequence		
MART + Ingredients (MART-I)	fry the onion rings in **oil** (✗ shrimp)	remove the **shrimp** from the **oil**
V + Ingredients (VI)	heat **oil** in a pan and add the **shrimp** and fry	remove the **shrimp** from the **oil**
VI + Visual simulator (VIV)	heat **oil** in a pan and fry the **shrimp** in it	remove the **shrimp** from the **oil**
VIV + Textual re-simulator (VIVT)	fry the **shrimp** in **oil**	remove the **shrimp** from the **oil**
Ground truth	place the **shrimp** into a pan of hot **oil**	remove the **shrimp** from the pan

Fig. 5. An example of generated procedural text. Green bold and red bold words represent semantically correct and incorrect ingredients, respectively. Words in parentheses indicate missing ingredients that should be included in the sentence. Note that parallel words in a sentence are not comma-separated due to the format of the YouCook2 dataset (cited from [20]).

Fig. 5 shows an example of generated recipes. Although the texts generated by our full model (VIVT) are not perfectly identical with the ground truth, the generated text still agrees with the content of each video segment.

Fig. 6 visualizes two examples of state transitions caused by action "beat" on materials "flour" and "egg". Each raw ingredients (red) transit to other points (blue), where the two nearest samples are video segments from other recipes but with similar contents. To further confirm the realization of state transition in a continuous latent feature space, we demonstrated some arithmetic operations with latent features (Fig. 7). Here, the right-hand video segments are retrieved on the 2D t-SNE space in Fig. 6. Although it can fail in some cases, we observed that we could explicitly simulate the transition by simple operations.

5 Conclusion

This paper introduced our recent attempts to model two major properties of manufacturing activities for cross-modal activity understanding. The first property is material merging processes, which we can model as a tree structure. Explicitly modeling the structure in the process of the visual storytelling task, we have enhanced captioning quality with any baseline methods. The second prop-

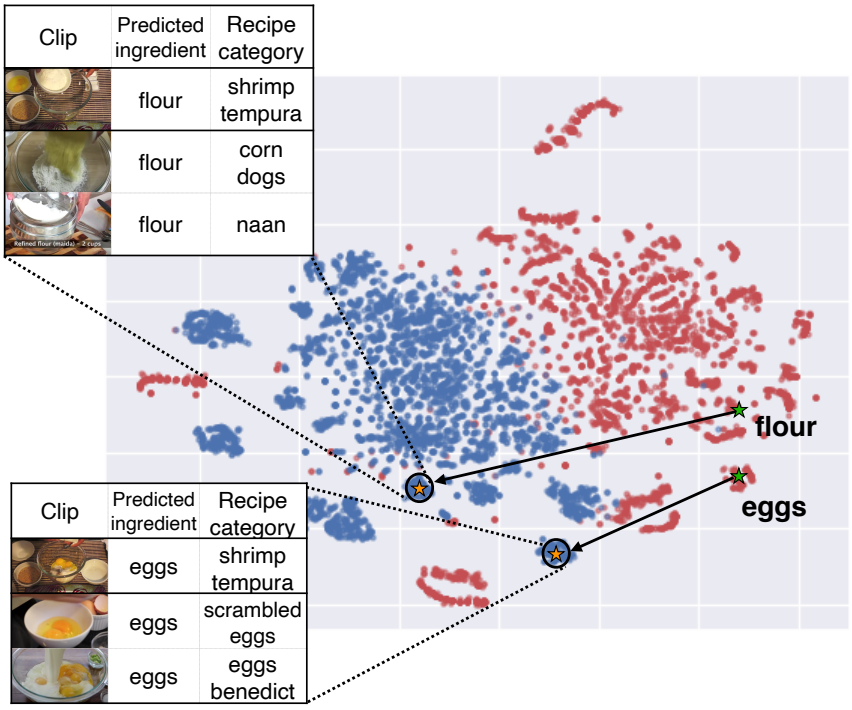


Fig. 6. A visualization of state transition in continuous latent space with tSNE [16]. Red and blue points represent raw and processed (updated) ingredient features, respectively (cited from [20]).

erty is the state transition of materials, which we can model as an action-driven transition in continuous space, using NPN. After the training, we confirmed that we had obtained a cross-modal representation that explicitly simulates changes caused by actions.

At the same time, we still have several future works. First, we should develop a method to model both properties simultaneously. Second, we are currently assuming preliminary segmented videos as input; however, this assumption seriously harms the method's usability. Therefore, a model that works with unsegmented videos is crucial to developing a commercial system. Finally, we need to roll out these techniques to applications other than cooking. The current methods lie on the large-size datasets, which are not always available as cooking recipe websites. We are starting to collect data on biochemical experiments [22], but it is not realistic to assume the same scale of a dataset for such tasks. Thus, we need to transfer knowledge from cooking tasks to others.

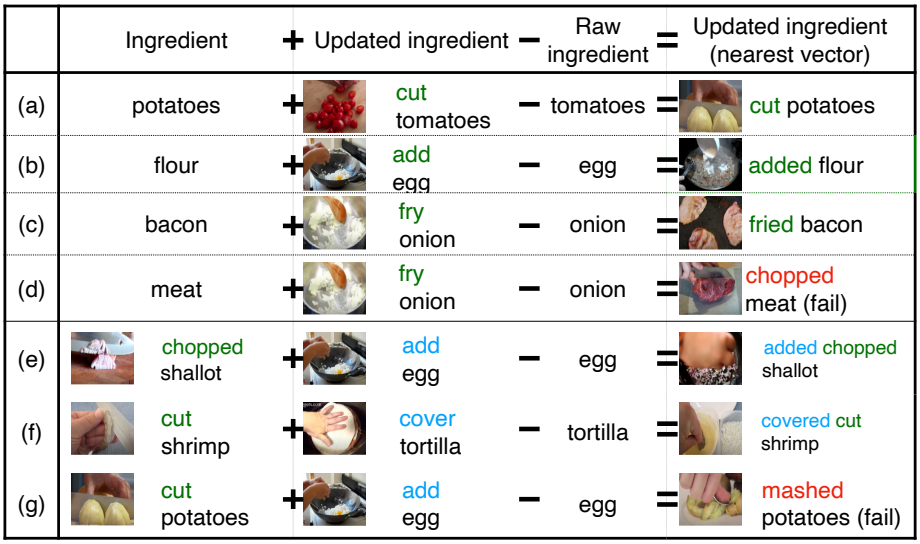


Fig. 7. Arithmetics using the learned latent features of ingredients. Examples (a) to (d) represent raw-to-processed transition, and (e) to (g) are processed-to-processed transition. Note that (d) and (g) shows failure cases (cited from [20]).

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP21J20250 and JP20H04210, and partially supported by JP21H04910, JP17H06100, JST-Mirai Program Grant Number JPMJMI21G2.

References

1. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proc. ACL Workshop IEEMMTS. pp. 65–72 (2005)
2. Bosselut, A., Levy, O., Holtzman, A., Ennis, C., Fox, D., Choi, Y.: Simulating action dynamics with neural process networks. In: Proc. ICLR (2018)
3. Chandu, K., Nyberg, E., Black, A.W.: Storyboarding of recipes: Grounded contextual generation. In: Proc. ACL. pp. 6040–6046 (2019)
4. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., Salakhutdinov, R.: Transformer-XL: attentive language models beyond a fixed-length context. In: Proc. ACL. pp. 2978–2988 (2019)
5. Gu, J., Im, D.J., Li., V.O.: Neural machine translation with Gumbel-greedy decoding. In: Proc. AAAI. pp. 5125–5132 (2018)
6. Harashima, J., Someya, Y., Kikuta, Y.: Cookpad image dataset: An image collection as infrastructure for food research. In: SIGIR (2017)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR. pp. 770–778 (2016)

8. Huang, T.K., Ferraro, F., Mostafazadeh, N., Misra, I., Agrawal, A., Devlin, J., Girshick, R., He, X., Kohli, P., Batra, D., Zitnick, C.L., Parikh, D., Vanderwende, L., Galley, M., Mitchell, M.: Visual storytelling. In: Proc. NAACL-HLT. pp. 1233–1239 (2016)
9. Jermsurawong, J., Habash, N.: Predicting the structure of cooking recipes. In: Proc. EMNLP (2015)
10. Kiddon, C., Ponnuraj, G.T., Zettlemoyer, L., Choi, Y.: Mise en place: Unsupervised interpretation of instructional recipes. In: EMNLP (2015)
11. Kim, T., Heo, M., Son, S., Park, K., Zhang, B.: GLAC net: glocal attention cascading networks for multi-image cued story generation. arXiv (2018)
12. Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proc. EMNLP. pp. 388–395 (2004)
13. Lei, J., Wang, L., Shen, Y., Yu, D., Berg, T., Bansal, M.: MART: memory-augmented recurrent transformer for coherent video paragraph captioning. In: Proc. ACL. pp. 2603–2614 (2020)
14. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: Proc. NAACL-HLT. pp. 110–119 (2016)
15. Lin, C.Y., Och, F.J.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Proc. ACL. pp. 605–612 (2004)
16. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
17. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: HowTo100M: learning a text-video embedding by watching hundred million narrated video clips. In: Proc. ICCV. pp. 2630–2640 (2019)
18. Mori, S., Maeta, H., Yamakata, Y., Sasada, T.: Flow graph corpus from recipe texts. In: Proc. LREC (2014)
19. Nishimura, T., Hashimoto, A., Mori, S.: Procedural text generation from a photo sequence. In: Proc. INLG. pp. 409–414 (2019)
20. Nishimura, T., Hashimoto, A., Ushiku, Y., Kameko, H., Mori, S.: State-aware video procedural captioning. In: ACMMM. pp. 1766–1774 (2021)
21. Nishimura, T., Hashimoto, A., Ushiku, Y., Kameko, H., Yamakata, Y., Mori, S.: Structure-aware procedural text generation from an image sequence. IEEE Access **9**, 2125–2141 (2020)
22. Nishimura, T., Sakoda, K., Hashimoto, A., Ushiku, Y., Tanaka, N., Ono, F., Kameko, H., Mori, S.: Egocentric biochemical video-and-language dataset. In: Proc. ICCVW. pp. 3122–3126 (2021)
23. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A method for automatic evaluation of machine translation. In: Proc. ACL. pp. 311–318 (2002)
24. Park, J.S., Rohrbach, M., Darrell, T., Rohrbach, A.: Adversarial inference for multi-sentence video description. In: Proc. CVPR. pp. 6598–6608 (2019)
25. Shi, B., Ji, L., Liang, Y., Duan, N., Chen, P., Niu, Z., Zhou, M.: Dense procedure captioning in narrated instructional videos. In: Proc. ACL. pp. 6382–6391 (2019)
26. Shi, B., Ji, L., Niu, Z., Duan, N., Zhou, M., Chen, X.: Learning semantic concepts and temporal alignment for narrated video procedural captioning. In: Proc. ACMMM. pp. 4355–4363 (2020)
27. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proc. ACL-IJCNLP. pp. 1556–1566 (2015)

28. Vedantam, R., Zitnick, C.L., Parikh, D.: CIDEr: Consensus-based image description evaluation. In: Proc. CVPR. pp. 4566–4575 (2015)
29. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
30. Wang, W., Wang, Y., Chen, S., Jin, Q.: YouMakeup: A large-scale domain-specific multimodal dataset for fine-grained semantic comprehension. In: Proc. EMNLP-IJCNLP). pp. 5133–5143 (2019)
31. Xiong, Y., Dai, B., Lin, D.: Move forward and tell: a progressive generator of video descriptions. In: Proc. ECCV. pp. 489–505 (2018)
32. Yagcioglu, S., Erdem, A., Erdem, E., Ikizler-Cinbis, N.: RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1358–1368 (2018)
33. Yamakata, Y., Mori, S., Carroll, J.: English recipe flow graph corpus. In: Proc. LREC. pp. 5187–5194 (2020)
34. Zhou, L., Xu, C., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. In: Proc. AAAI. pp. 7590–7598 (2018)
35. Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)