

Visual Grounding Annotation of Recipe Flow Graph

Taichi Nishimura¹, Suzushi Tomori¹, Hayato Hashimoto¹, Atsushi Hashimoto²,
Yoko Yamakata³, Jun Harashima⁴, Yoshitaka Ushiku², Shinsuke Mori⁵

¹Graduate School of Informatics, Kyoto University, ²OMRON SINIC X Corporation

³Graduate School of Information Science and Technology, The University of Tokyo

⁴Cookpad Inc, ⁵Academic Center for Computing and Media Studies, Kyoto University

^{1,5}Yoshidahonmachi, Sakyo-ku, Kyoto, Japan, 606-8501

²5-24-5, Hongo, Bunkyo-ku, Tokyo, Japan, 113-0033

³7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan, 113-8656

⁴4-20-3 Ebisu, Shibuya-ku, Tokyo, Japan, 150-6012

¹{nishimura.taichi.43x,tomori.suzushi.72e,hashimoto.hayato.73e}@st.kyoto-u.ac.jp,

²{atsushi.hashimoto, yoshitaka.ushiku}@sinicx.com,

³yamakata@mi.u-tokyo.ac.jp, ⁴jun-harashima@cookpad.com, ⁵forest@i.kyoto-u.ac.jp

Abstract

In this paper, we provide a dataset that gives visual grounding annotations to recipe flow graphs. A recipe flow graph is a representation of the cooking workflow, which is designed to understand the workflow from natural language processing. Such a workflow will increase its value when grounded to real-world activities, and visual grounding is a way to do so. Visual grounding is provided as bounding boxes to image sequences of recipes, and each bounding box is linked to an element of the workflow. Because the workflows are also linked to the text, this annotation gives visual grounding with workflow’s contextual information between procedural text and visual observation in an indirect manner. We subsidiarily annotated two types of event attributes with each bounding box: “doing-the-action,” or “done-the-action”. As a result of the annotation, we got 2,300 bounding boxes in 272 flow graph recipes. Various experiments showed that the proposed dataset enables us to estimate contextual information described in recipe flow graphs from an image sequence.

Keywords: Procedural Text, Bounding Box, Flow Graph, Visual Grounding

1 Introduction

Procedural texts are suitable for the target of natural language understanding (NLU) because they are goal-oriented descriptions and we can almost define the understanding of their goals. Some studies have proposed the framework of understanding procedural texts with graphs, which can represent the entire workflow (Mori et al., 2014; Jermurawong and Habash, 2014; Kiddon et al., 2015). These frameworks would be helpful for real-world systems, such as smart kitchen (Hashimoto et al., 2008) and cooking robot (Bollini et al., 2013), to understand the context of these workflows and execute next actions. Therefore, it is essential to ground visual observations (images or videos) with procedural texts.

Visual grounding is one of the solutions to help computers to understand which objects are aligned with textual descriptions. In the previous studies, they targetted the pair of general texts and visual observations. MSCOCO (Lin et al., 2014), Flickr30k (Plummer et al., 2015), and YouTube-BoundingBox (Real et al., 2017) are typical datasets for such tasks. They reported that annotating bounding boxes with a textual description helps computers to know which objects they have to pay attention to for image captioning (Xu et al., 2015; Cornia et al., 2019), visual question answering (VQA) (Fukui et al., 2016), and some other visual grounding tasks (Huang et al., 2017; Bojanowski et al., 2015; Zhang and Lu, 2018). Compared with such general visual grounding tasks, a dataset with procedural texts is differentiated by an additional goal of understanding the context information, i.e., a model should consider the entire workflow of a procedural text. We call

this task contextual visual grounding.

In this background, this paper focuses on the domain of cooking recipes and provides a new dataset that has visual grounding annotation of contextual information represented in recipe flow graph (r-FG) (Mori et al., 2014), which can represent a structured workflow described in a procedural text. Figure 1 shows an overview of our annotation scheme. Given a procedural text with its image sequence and flow graph, annotators put a bounding box of an object in an image, which is linked to a node of the flow graph. This annotation enables us to link visual objects to contextual information directly. Through the process, as a subsidiary contribution, we found that visual objects represent roughly in either of the following two events: “doing-” and “done-the-action.” In the dataset, we also provide attribute labels for these events. For example, in the “step 1” in Figure 1, the chef is just dicing the broccoli, whereas the chef has done the dicing action in the “step 2” of Figure 1. To make it obvious whether an action to a visual object is in performance or has already been performed, we annotated each bounding box with an event attribute.

As a result of the annotation, we built an r-FG bounding box (r-FG-BB) dataset, which consists of 2,300 bounding boxes in 272 r-FGs. In the experiments, we confirmed that, with our dataset, we could obtain a model that grounds bounding boxes with nodes in a flow graph, which gives a connection between the bounding boxes of images, while classifying event attributes correctly.

Our dataset allows contextual visual grounding research. Grounded caption generation is a typical usage of our dataset, where the pairs of a node in the flow graph

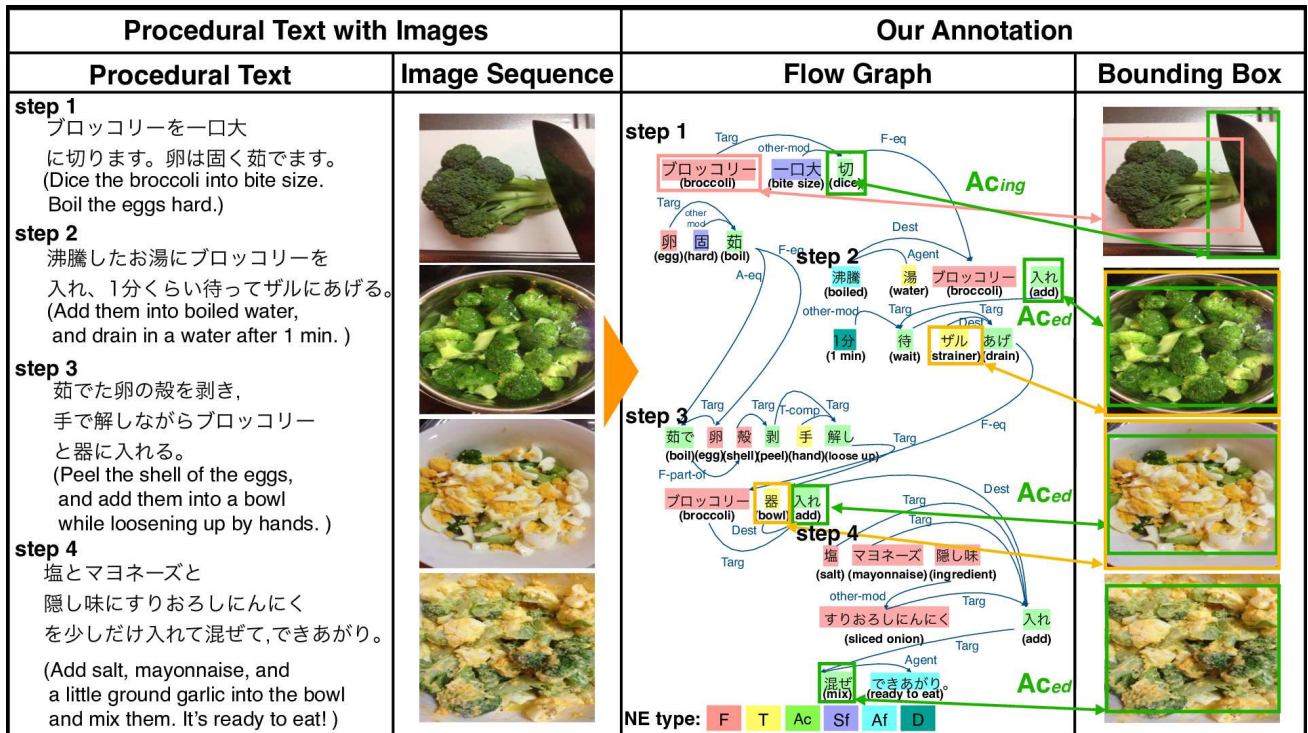


Figure 1: An overview of our annotation.

Concept tag	Meaning	Frequency
F	Food	12.37
T	Tool	3.59
D	Duration	0.73
Q	Quantity	0.74
Ac	Action by the chef	13.21
Af	Action by foods	2.67
Sf	State of foods	3.03
St	State of tools	0.31
Total	—	36.65

Table 1: Recipe named entity (r-NE) tags and frequencies per recipe in our r-FG-BB dataset.

and bounding boxes help a model understand which region they have to pay attention to at decoding a sentence. One may leverage the pairs for multimodal state tracking (Yagcioglu et al., 2018; Amac et al., 2019) by tracing nodes via arcs in the flow graphs.

2 Recipe Flow Graph Corpus

The language part of our r-FG-BB dataset is compatible with the r-FG corpus (Mori et al., 2014). The r-FG corpus consists of cooking recipe texts (or simply “recipes” hereafter) annotated with flow graphs connecting important terms in the texts. Our r-FG-BB dataset is an extension connecting some of the important terms to bounding boxes in the image attached to each step in the text. In this section, we explain cooking recipe texts and flow graphs of the r-FG corpus.

2.1 Cooking Recipe Text

A recipe describes instructions for a dish. The left part of Figure 1 shows an example in Japanese with its English translation. A recipe has a sequence of steps, and a step consists of instruction sentences. The recipes in the r-FG corpus are randomly selected from a dataset consisting of recipes downloaded from the recipe hosting service Cookpad¹. In addition to texts, Cookpad allows users to attach an image to each step to explain it better (see the image sequence column in Figure 1 for example). In our r-FG-BB dataset, we capitalize on these characteristics.

2.2 Directed Acyclic Graph Representation

In a recipe, the order of the instructions is not fully but only partially specified. For example, a chef needs to dice carrots before boiling them but does not need to dice potatoes to boil carrots. In this case, the action “dicing carrots” has an order relationship with the action “boiling carrots,” but not with the action “dicing potatoes.” Moreover, the diced carrots and boiled carrots are identical. A directed acyclic graph (DAG) is suitable for representing such relationships. The r-FG is a DAG whose nodes correspond to important terms (word sequences) with a type in the recipe. The arcs represent the relationships between two nodes. All the nodes are connected indirectly to a single special node, root, corresponding to the final dish. Thus a meaning of a recipe in the r-FG corpus is represented as a rooted DAG.

2.2.1 Nodes

The nodes of an r-FG are the important terms appearing in the sentences with a type. Table 1 lists the types and their

¹<https://cookpad.com/> (Accessed on 2019/Nov/25)

Arc label	Meaning	Frequency
Agent	Action agent	2.58
Targ	Action target	14.50
Dest	Action destination	5.65
F-comp	Food complement	0.44
T-comp	Tool complement	1.31
F-eq	Food equality	2.59
F-part-of	Food part-of	3.25
F-set	Food set	0.26
T-eq	Tool equality	0.23
T-part-of	Tool part-of	0.45
A-eq	Action equality	0.58
V-tm	Head of a clause for timing	1.15
other-mod	Other relationships	5.23
Total	–	38.21

Table 2: Arc labels and their frequencies per recipe in our r-FG-BB dataset.

frequencies per a recipe in the r-FG-BB dataset. An important term is a sequence of words annotated with a type among a pre-defined set. This definition is similar to that of named entities (NE) except for the type set. Thus in the r-FG corpus, it is called recipe NE (r-NE). Each node of a flow graph corresponds to an r-NE in the text. As we described above, there is a special node, root, corresponding to the completed dish. In Figure 1, the root is the node whose r-NE is “ready to eat/Af.”

Our r-FG-BB dataset extends the r-FG corpus by annotating bounding boxes in the images. We assume that each step in the recipes of our dataset has an image. We annotate some objects in the image with a bounding box and a link to the r-NE in the corresponding step. As a first trial we limited the r-NE types into F (Food), T (Tool), and AC (Action by the chef) among eight types listed in Table 1. The reason why we excluded the others is that they are difficult or almost impossible to be recognized from an image. In addition, the statistics in Table 1 indicate that the three types occupy approximately 79.6%. Thus we decided to take these three visually obvious types, F, T, and AC, leaving others for a future challenge.

2.2.2 Arcs

An arc between two nodes indicates that they have a certain relationship. The type of relationship is denoted by a label of the arc. Table 2 lists the arc labels and their frequencies per a recipe in our r-FG-BB dataset. Unlike an arc in a dependency parsing, an arc in an r-FG may connect two nodes in different sentences. For example, in Figure 1, the r-NE node “boil” in step 1 and “eggs” in step 3 are connected by the F-eq arc indicating that the eggs in step 3 are identical to the result of the action “boil the eggs hard”.

In our r-FG-BB dataset, all the bounding boxes are connected directly or indirectly by arcs in the r-FG. This characteristic distinguishes our r-FG-BB dataset from other existing visual datasets.

3 Dataset Annotation Standard

In this section, we explain our annotation framework in detail. As mentioned in Section 1, we performed two types of







F (Food)	T (Tool)	Ac (Action by the chef)
 <p>Prepare two eggs.</p>	 <p>Add the tuna and seasoning to the bowl.</p>	<p>doing (AC_{ing})</p>  <p>Dice the broccoli into bite size.</p>
 <p>recommend cotton tofu.</p>	 <p>Put miso in the fried tofu using a spoon.</p>	<p>done (AC_{ed})</p>  <p>Cut the spinach into bite size pieces.</p>

Figure 2: Annotation examples.

annotations: bounding boxes with nodes in the flow graph and event attributes with bounding boxes. We describe these annotations in the subsequent sections.

3.1 Bounding Box Annotation

Similar to other visually annotated datasets, we annotate some objects in images with a bounding box. The edges of a bounding box are parallel to the vertical or horizontal edge of the image (assumed to be a rectangle). A bounding box, specified by two corner points, is the smallest area covering the target object completely. In our r-FG-BB, we limited the target objects into those having a corresponding r-NE in the flow graph, that had been manually annotated to a recipe. We also annotate each bounding box with a link to the corresponding r-NE. And some bounding boxes may be linked to the same r-NE because the object can be multiple or separated in the image. Thus the relationship from bounding boxes to r-NEs is so-called one-to-many. These links are the source of all the interesting points of this work. For example, there are two bounding boxes contouring the eggs in the image at the top left of Figure 2 and they are connected to a single r-NE “eggs.” By the linguistic annotation in the r-FG part, we can say many things about the relationships of this object with others.

A node of type F has a tendency to be linked to bounding boxes for unprocessed ingredients (see Figure 2). Here, “unprocessed ingredient” indicates a food item to which the chef has not performed any action yet (thus diced potatoes and grilled chicken is not “unprocessed”). In some cases, processed food is specified by a noun phrase corresponding to some bounding boxes. A node of type T should be linked to the bounding box of a tool. We also annotate tools appearing only partially in an image with a bounding box spreading to the edge of the image. Different from F and T, the bounding box of an AC is not defined in a straightforward way. It can be a bounding box of the tool to be used or that of the processed food, which shows the result of the action AC.

3.2 Event Attribute Annotation

In addition to the link between an AC and a bounding box, we annotate the link with an attribute indicating that the



Figure 3: Annotation screen of developed web tool.

image shows an on-going action (Ac_{ing}) or a completed one (Ac_{ed}). Note that this attribute is only for Ac . We call this an event attribute. For example, at the top right in Figure 2, the link between the action “Dice” in the text below and the blue bounding box should have an event attribute Ac_{ing} indicating that the action is on-going. And at the bottom right, the link between the action “Cut” in the text below and the blue bounding box would have the attribute Ac_{ed} that indicates that the action is completed. In some cases, it is unclear whether the action is on-going or completed. We allow an annotator to label such a case with Ac_{unc} .

3.3 Annotation Web Tool

To facilitate annotation, we developed a web tool. Figure 3 shows a screenshot. On the left-hand side, the tool shows an image corresponding to a step. On the right-hand side, the tool shows all the steps of a recipe with all F, T, and Ac nodes highlighted with a color depending on the type.

The annotator checks all r-NEs from the first to the last in the step whose image is shown on the left-hand side whether the image shows the corresponding object of the focused r-NE or not. If it shows the object, the annotator draws its bounding box in the image. The tool allows the annotator to adjust the box by moving or rescaling it. The tool also has a function to delete a bounding box.

After annotating a bounding box, the annotator connects it with a link to the node in the flow graph. If the node is an Ac , the annotator is forced to select one event attribute from three categories (Ac_{ed} , Ac_{ing} , and Ac_{unc}) listed in the pull-down menu.

A typical case for Ac_{unc} is an instruction “/Add/Ac salt to boiling water” with an image of boiling water only. From the image, it is almost impossible to judge if the salt has been added or not yet. The label Ac_{unc} allows the annotator to avoid making a difficult decision.

4 Annotation Statistics

As the recipes for our r-FG-BB dataset, we selected, from the r-FG corpus, those having an image to each step. As a result, we had 70 recipes. To increase the number of recipes in our corpus, we also selected 202 recipes randomly from those consisting of 3 to 10 steps in the Cookpad Image

#Steps	#Sent.	#r-NEs		#Words	#Char.
		#Leaves	#Non-Leaves		
4.97	7.34	13.48	36.65	113.99	173.91
			23.16		

Table 3: Per-recipe statistics of our r-FG-BB corpus of 272 recipes.

	F	T	Ac	Total
Agreement rate	0.81	0.88	0.76	0.79

Table 4: Agreement rate between two annotators

Dataset (Harashima et al., 2017) under the same condition. The r-FG for each recipe was annotated carefully by the same annotators as the r-FG corpus. Combining the above two sources, we obtained 272 recipes and their r-FG annotations. Table 3 shows the statistics of the r-FG part of our r-FG-BB dataset. Then two annotators drew bounding boxes in the images of these recipes and connected them to r-NEs.

In this section, we first report the agreement rates between the two annotators decomposing the annotation into bounding boxes, selections of r-NEs connected to them, and event attributes. Then we describe the procedure we have executed to build our r-FG-BB dataset considering the agreement rate. Finally, we show its statistics.

4.1 Agreement Rate

We calculated the agreement rates of r-NEs and bounding boxes (selection problems) like other image datasets. The agreement rate is defined as the intersection over union (IoU) calculated by the following formula:

$$IoU(S_A, S_B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}, \quad (1)$$

where S_A and S_B are the sets of annotations by the annotator A and B , respectively.

For an evaluation of event attribute annotation (a classification problem), we calculated the confusion matrix.

4.1.1 r-NE

In the r-NE case, an element of the set S is an r-NE selected to connect to bounding boxes. We calculated the agreement rate of selected r-NEs between two annotators for each type and in total.

Table 4 shows the result. We see that the agreement rate is highest for T and lowest for Ac. The reason is that tools (T) are solid and do not change the shape, but the object corresponding to an Ac tends to be processed food, which is sometimes difficult to be identified completely. We see that the total IoU of 79% can be seen relatively high to build a dataset.

4.1.2 Bounding Box

In the case of bounding boxes annotation, an element of the set S is an area specified by the bounding box. We first calculated the pixel-wise IoU for each pair of the bounding boxes annotated by the two annotators. We limited the bounding boxes to those connected to an r-NE selected by both the annotators. In some cases, an annotator gives more

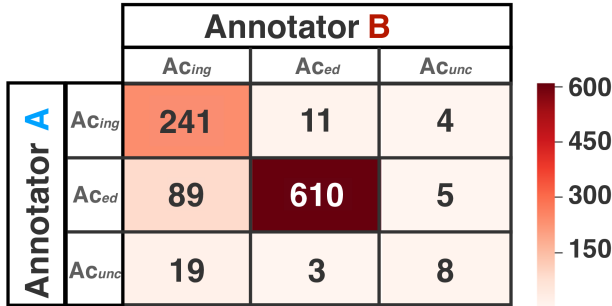


Figure 4: Event attribute confusion matrix between two annotators.

	F	T	Ac (AC _{ing} , AC _{ed})	Total
#r-NEs	0.16	1.28	2.44 (0.62, 1.82)	3.88
#BBs/#r-NEs	2.00	2.00	2.28 (2.59, 2.18)	2.18

Table 5: Numbers of r-NEs connected to bounding boxes (BBs) per recipe and numbers of bounding boxes per r-NE.

than one bounding box corresponding to the same r-NE. Thus we calculated IoU scores of all the combinations of the bounding boxes connected to the same r-NE. Then we calculate the average IoU.

The average IoU was 0.772. In the computer vision community, researchers build their datasets with bounding boxes that have IoUs larger than 0.5 (Su et al., 2012; Papadopoulos et al., 2017). Therefore, we can say that many annotated bounding boxes are suitable for the dataset.

4.1.3 Event Attribute

As we described, if the connected r-NE type is AC, the annotator chooses one event attribute among AC_{ing}, AC_{ed}, and AC_{unc}. We limited the ACs to those selected by both the annotators. Since the numbers of total annotations by the two annotators are the same, we calculated the confusion matrix. Figure 4 shows the result. As we see from the table, 87% (= 859/990) of the event attributes agreed. In terms of AC_{unc}, there are very few tags selected by each annotator. The most frequent attribute that two annotators agreed is AC_{ed}. This may be because the recipe authors need to clean up their hands to use the camera and from processed foods, it is obvious that the action has been completed.

4.2 Dataset Construction

After the annotation, we applied the following two filtering processes to construct our r-FG-BB dataset.

1. We selected pairs of an r-NE and bounding boxes from the annotation result only when the two annotators are in agreement. As a result, 79.3% of the annotated r-NEs are extracted as shown in Table 4. Then, we filtered out the pairs of an AC and bounding boxes when its event attribute is AC_{unc}.
2. We further filtered out pairs of an r-NE and bounding boxes if the IoU of the bounding boxes is over 0.7.

Finally, we obtained 2,300 pairs of an r-NE and bounding boxes for our r-NE-BB dataset consisting of 272 recipes.

Table 5 shows the numbers of r-NEs connected to bounding boxes per recipe and its average numbers of the bounding boxes. The bounding boxes from the two annotators are counted independently. Hence, some regions are double-counted. In this table, the numbers of AC_{ing} and AC_{ed} are counted only when the two annotators are in agreement. We see that the number of F is very small because raw ingredients are rarely photographed.

5 Experimental Evaluation

To investigate the usefulness of our dataset, we propose three tasks based on its most interesting characteristics: symbol grounding, bounding box linking prediction, and event attribute classification. In this section, we define these tasks, give a preliminary solution to each one, and report the experimental results.

5.1 Symbol Grounding

5.1.1 Task Definition

Contextual visual grounding is a visual grounding task that requires a model to understand contextual information described in a procedural text. Given a bounding box in an image and the corresponding step to the image, the task is to select an r-NE (F, T, or AC) in the step. Formally, let a step be $W = (w_1, w_2, \dots, w_n, \dots, w_N)$, where w_n is the n -th word and N is the number of words in the step, and the r-NEs in the step be $R = (r_1, r_2, \dots, r_k, \dots, r_K)$, where K is the number of r-NEs in the step. The goal of this task is to select one r-NE \hat{r}_k among R given a bounding box and the word sequence of the step corresponding to the image W . To investigate whether the flow graph is effective or not, we allow some models to refer to the flow graph of the entire recipe.

5.1.2 Proposed Model

To build a model for the symbol grounding task, we have to consider that the step W contains each r-NEs R . Thus we adopted an embedding-based approach, which enables us to calculate the similarity between all combinations of the bounding box and candidate r-NEs. The r-NE that has the highest score is selected to be the answer.

Figure 5 shows an overview of our approach. (i) We convert the bounding box and candidate r-NEs into feature vectors. For the bounding box, the image encoder calculates the image vector v . Similarly for the corresponding step W the textual encoder outputs the sequence of word embedding vectors $W = (w_1, w_2, \dots, w_n, \dots, w_N)$. Then, we calculate the embedding vector of the k -th r-NE as the average of the vectors of the words consisting that r-NE: $R = (r_1, r_2, \dots, r_k, \dots, r_K)$. (ii) For models referring to the flow graph, we concatenate two vectors to r_k . One is the preceding context vector r_k^p calculated as the average of the vectors of all the r-NEs in-coming to the k -th r-NE in the flow graph. The other is the following context vector r_k^f calculated as the average of the vectors of all the r-NEs to which the k -th r-NE is out-going in the flow graph. If the k -th r-NE is a leaf or the root, we set random values to all the elements of r_k^p or r_k^f , respectively. These vectors are expected to represent the context information. Thus, the enhanced r-NE vectors are given

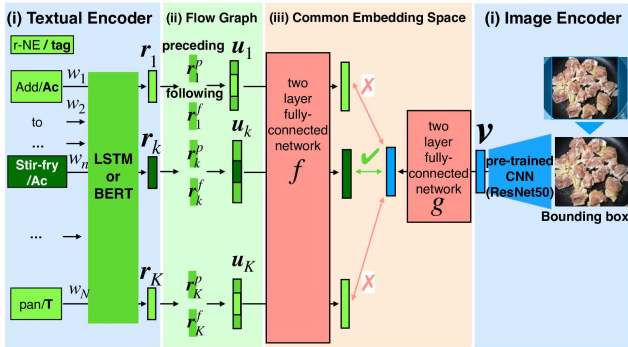


Figure 5: An overview of our symbol grounding model. We used the pre-trained neural networks for the image and textual encoder.

random baseline	Recall	Precision	F1
F	0.023	0.250	0.042
T	0.091	0.039	0.054
Ac	0.429	0.221	0.291
Total	0.189	0.174	0.181
LSTM w/o flow graph			
F	0.250	0.250	0.250
T	0.480	0.461	0.471
Ac	0.634	0.662	0.648
Total	0.580	0.592	0.586
LSTM w/ flow graph			
F	0.125	0.250	0.167
T	0.772	0.654	0.708
Ac	0.612	0.603	0.607
Total	0.608	0.602	0.605
BERT w/o flow graph			
F	0.000	0.000	0.000
T	0.720	0.750	0.734
Ac	0.762	0.716	0.739
Total	0.733	0.717	0.725
BERT w/ flow graph			
F	0.000	0.000	0.000
T	0.782	0.750	0.766
Ac	0.785	0.761	0.772
Total	0.734	0.750	0.742

Table 6: Result of the symbol grounding experiment.

as $u_k = \text{concat}(r_k^p, r_f, r_k^h)$, where $\text{concat}(\cdot)$ is the vector concatenation function. When we do not use the flow graph representation, we skipped this phase. Thus r-NE vectors are given as $u_k = r_k$. (iii) We trained the two neural networks, f from the textual branch and g from the image branch, jointly to calculate the common mapping space from the r-NE vectors and the image vectors. The goal is to train these networks to embed a given r-NE/bounding box pair at closer positions while others in the distance. To achieve this, we used the triplet margin loss (Balntas et al., 2016) with the distance function as the cosine distance.

5.1.3 Results

To perform this experiment, we used all the pairs of an r-NE and a bounding box in our r-FG-BB dataset. We

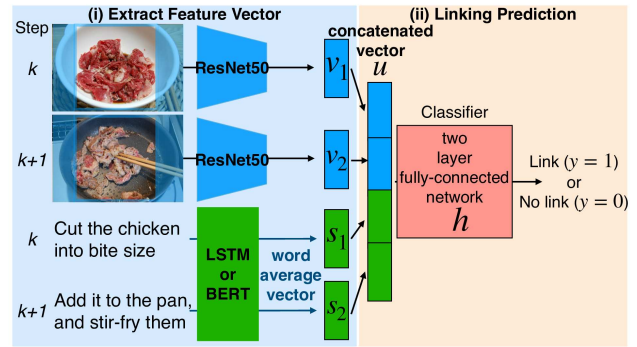


Figure 6: An outline of our bounding box linking classifier. The image encoder and textual encoder are the same models to Section 5.1.

split the dataset into 80% for training, 10% for validation, and 10% for test. As the image encoder, we used ResNet-50 (He and Sun, 2016) pre-trained with ImageNet (Deng et al., 2009), which is one of the state-of-the-art image classifiers. We used the classifier after removing the last layer with softmax; thus, the output layer dimension is 2048. For the textual encoder, we prepared two choices: LSTM and BERT (Devlin et al., 2019). We trained them using 0.5 million recipes from the Cookpad Dataset (Harashima et al., 2016). For training LSTM, we used the tokenizer KyTea² (Neubig et al., 2011). We replaced words appearing less than three times with an unknown word symbol to have a 17,982-word vocabulary for the LSTM. The perplexity was 29.16. For the BERT-based model, we employed a pre-trained BERT model on the Wikipedia corpus. Then, we fine-tuned the model with recipes in the same manner as the LSTM training. The vocabulary size of BERT is 32,000 after converting words into sub-word units (Sennrich et al., 2016). Table 6 shows the results. Independently from the textual encoder, methods referring to the flow graph performed better than those without flow graph reference. This suggests that the context information from the flow graph helps them model link a bounding box in an image and a named entity in the corresponding text. We also see that BERT-based models performed better than those based on LSTM. This is consistent with other tasks in natural language processing.

5.2 Bounding Box Linking Prediction

5.2.1 Task Definition

In our r-FG-BB dataset, some bounding boxes are connected through the flow graph. Thus we can try an interesting novel task: bounding box linking. We defined the task as a classification task to predict whether there is a link or not between a given pair of bounding boxes in two subsequent bounding boxes.

5.2.2 Proposed Model

Figure 6 shows an overview of our proposed approach. The inputs are two bounding boxes in subsequent images corresponding to the k -th and $(k + 1)$ -th steps. (i) Similar to

²<http://www.phontron.com/kytea/>

Baseline	Accuracy
Random	0.483
Cosine similarity ($\alpha = 0.8$)	0.533
Proposed method	
Image	0.667
Image + LSTM	0.583
Image + BERT	0.817

Table 7: Accuracies of bounding box linking classification.

the previous task, we converted the bounding boxes into the image vectors, v_1 and v_2 . Then we calculated the step vectors, s_1 and s_2 , as the average vectors of the words in the step. (ii) We concatenated them and obtain the feature vector $u = \text{concat}(v_1, v_2, s_1, s_2)$. Finally, the feature vector u is fed to a classifier h , which is a fully-connected two-layer network followed by a sigmoid activator. In the classification phase we take argmax to output whether there is a link ($y = 1$) or not ($y = 0$). In the training phase, we minimized the binary cross-entropy loss with the Adam optimizer (Kingma and Ba, 2015).

5.2.3 Results

To perform this experiment, first, we split the r-FG-BB dataset into 80% for training, 10% for validation, and 10% for test. Then we extracted pairs of bounding boxes from each of two subsequent steps. Finally we checked whether the two bounding boxes have a link or not in the flow graph to prepare the label ($y = 1$ or 0).

In this experiment, we employed two baselines: random and cosine similarity. The random baseline investigates whether the link label is biased or not. In ‘‘cosine similarity,’’ the model calculates the cosine similarity between the image vectors v_1 and v_2 and judges that there is a link if the score is higher than a threshold α , which were set to 0.8. Note that two image vectors are L2 normalized.

Table 7 shows the results of these two baselines and the proposed methods. The result of the random baseline shows that the task is a difficult binary classification problem without the model. The accuracy of the ‘‘cosine similarity’’ is slightly higher than that of the random. This indicates that this task is not so simple to be able to be answered correctly without textual information. Compared with the baselines, the proposed models could predict bounding box linking more correctly. In the proposed method, the model learned with only images achieved the highest precision, but that learned with texts using BERT in addition to the images achieved the highest recall and F1. In contrast, the result of the model using LSTM was even worse than that trained with only images. This indicates that a sophisticated textual representation is necessary to solve this task.

5.3 Event Attribute Classification

5.3.1 Task Definition

Each bounding box has two event attributes, ‘‘doing the action’’ and ‘‘done the action’’. It is important for a computer to be able to distinguish between these two from visual and textual information because real-world systems require a model to know the status of actions. Our dataset has an event attribute for each bounding box connected to an ac-

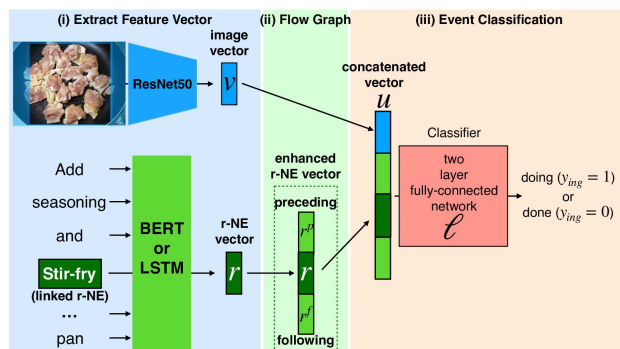


Figure 7: An overview of our event attribute classification model. The encoders are also the same model to the symbol grounding models described in Section 5.1.

	Accuracy
Image	0.750
+ LSTM w/o flow graph	0.839
+ LSTM w/ flow graph	0.893
+ BERT w/o flow graph	0.839
+ BERT w/ flow graph	0.875

Table 8: Accuracies of event attribute classification.

tion node AC . The labels are AC_{ing} for on-going action and AC_{ed} for a completed one. Thus our dataset allows us to test methods for this classification as well. We formulated this problem as a binary classification task. Given a bounding box connected to AC along with the step, we train models to classify whether the action is on-going or completed ($y = AC_{ing}$ or AC_{ed}).

5.3.2 Proposed Model

Figure 7 shows an outline of our event attribute classification model. (i) Through the encoders, we get the image embedding vector v and the r-NE vector r . (ii) If the model is allowed to refer to a flow graph, we concatenated the preceding and following context vectors r^p, r^f to the r in the same manner as Section 5.1. We get the feature vector by concatenating them: $u = \text{concat}(v, r^p, r, r^f)$. Then, we provide it to a fully-connected two-layer network followed by a sigmoid activator. The network outputs whether the event attribute is AC_{ing} or AC_{ed} . In the training phase, we searched for the parameters that minimize the binary cross-entropy loss by Adam optimizer. (Kingma and Ba, 2015).

5.3.3 Results

To perform this experiment, we extracted pairs of an AC and bounding box whose event attributes is AC_{ing} or AC_{ed} . Then, we split the data into 80% for training, 10% for validation, and 10% for test. Table 8 shows the results of the models. Before this experiment, we anticipated that the model could tell them apart only from image information because object appearance is significantly different between visual information annotated with AC_{ing} and AC_{ed} . The model could distinguish them to a certain extent (accuracy: 75%), but the result revealed that incorporating textual encoders such as BERT and LSTM contributed to a higher score. From this result, we can safely say that lin-

guistic information is effective for event attribute classification. Compared with the models without reference to the flow graph, the models with it perform better with both textual encoders, LSTM and BERT. This result indicates that the flow graph representation also helps the model to solve this task.

6 Application

Our r-FG-BB dataset has many potential applications. In this section, we describe two typical usages of our dataset.

6.1 Grounded Image Captioning

Grounded image captioning is an essential problem in the computer vision and natural language communities. In this field, to output grounded caption, many researchers have incorporated the attention mechanism into a model (Xu et al., 2015; Cornia et al., 2019). This attention mechanism encourages models to know which objects and regions they have to pay attention to when decoding words, and it helps them to generate visually grounded captions. Our dataset provides researchers with pairs of an r-NE and bounding boxes, which leads a model to decode visually grounded words correctly. Moreover, we would like to emphasize that the text part is not a general text but a procedural text. Procedural text generation from an image sequence is a prominent area because it requires a model to consider the context and the output coherency (Nishimura et al., 2019; Chandu et al., 2019). Thus these studies focused on generating grounded captions: incorporating a structure into the model implicitly (Chandu et al., 2019) and preferentially decoding important terms (Nishimura et al., 2019). Our dataset accelerates this research because pairs of an r-NE and bounding boxes help a model output these terms, considering the entire workflow using the flow graph.

6.2 Multimodal State Tracking

State tracking, which detects changes of an object and its identity in a text, is an essential problem for NLU. Recently, procedural texts are becoming the target for NLU researchers because their understanding requires a model to anticipate the implicit causal effects of actions on entities (Bosselut et al., 2018; Tandon et al., 2018; Mishra et al., 2018; Yagcioglu et al., 2018). Some researchers tried to build a dataset for procedural text understanding (Tandon et al., 2018) for the scientific domain, and other researchers expand them to multimodal version (Yagcioglu et al., 2018) for the cooking area. Some researcher has proposed a dataset for the understanding of procedural texts using a “grid,” which represents changes of an object state using a table format (Tandon et al., 2018; Mishra et al., 2018). Our dataset provides a flow graph of a procedural text, which represents an entire workflow as a rooted DAG, and bounding boxes in an image connected to nodes in the flow graph. Our experimental results showed that the model could predict which textual description is related to a bounding box and whether there exists a linking between bounding boxes or not. Thus, sequential bounding boxes and a flow graph will be useful for multimodal state tracking. It would be interesting to build a

QA dataset, which requires a model to understand not only flow graphs but also bounding boxes. For example, given an image showing mixing ingredients in a bowl, the question is, “What ingredients are used in it?”. To answer this question correctly, the model must understand the flow of all the ingredients and their visual locations.

7 Conclusion

In this paper, we presented details of our r-FG-BB dataset. Its language part is procedural texts, each of which is annotated with a flow graph whose nodes correspond to important terms in the text and are connected to form a DAG. The visual part of the dataset is bounding boxes in the images attached to each text. The bounding boxes are connected to the nodes of the flow graph, allowing the dataset users to have contextual information of the visual objects.

We performed various experiments. The results showed that the proposed simple models could ground bounding boxes with nodes in a flow graph, which gives a linking between the bounding boxes over images while classifying event attributes correctly. Therefore, our dataset is useful for contextual visual grounding.

With our dataset, one can try contextual visual grounding research using interesting triplets: bounding boxes, procedural text, and its flow graph. This combination would be effective for understanding the entire workflow of a procedural text, grounding contextual visual information.

Acknowledgement

This work was supported by JST ACT-I Grant Number JP-MJPR17U5.

References

- Amac, M. S., Yagcioglu, S., Erdem, A., and Erdem, E. (2019). Procedural reasoning networks for understanding multimodal procedures. In *Proceedings of the Conference on Computational Natural Language Learning*.
- Balntas, V., Riba, E., Ponsa, D., and Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference*, pages 1–11.
- Bojanowski, P., Lajugie, R., Grave, E., Bach, F., Laptev, I., Ponce, J., and Schmid, C. (2015). Weakly-supervised alignment of video with text. In *Proceedings of the International Conference on Computer Vision*, pages 4462–4470.
- Bollini, M., Tellex, S., Thompson, T., Roy, N., and Rus, D. (2013). Interpreting and executing recipes with a cooking robot. In *Experimental Robotics*, pages 481–495.
- Bosselut, A., Levy, O., Holtzman, A., Ennis, C., Fox, D., and Choi, Y. (2018). Simulating action dynamics with neural process networks. In *Proceedings of the International Conference on Learning Representations*.
- Chandu, K., Nyberg, E., and Black, A. W. (2019). Storyboarding of recipes: grounded contextual generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6040–6046.

- Cornia, M., Baraldi, L., and Cucchiara, R. (2019). Show, control and tell: a framework for generating controllable and grounded captions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 8307–8316.
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.
- Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M. (2016). Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 457–468.
- Harashima, J., Ariga, M., Murata, K., and Ioki, M. (2016). A large-scale recipe and meal data collection as infrastructure for food research. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 2455–2459.
- Harashima, J., Someya, Y., and Kikuta, Y. (2017). Cookbook image dataset: An image collection as infrastructure for food research. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1229–1232.
- Hashimoto, A., Mori, N., Funatomi, T., Yamakata, Y., Kakusho, K., and Minoh, M. (2008). Smart kitchen: a user centric cooking support system. In *Proceedings of the Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 848–854.
- He, K., Z. X. R. S. and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 770–778.
- Huang, D. A., Lim, J. J., Fei-Fei, L., and Carlos Niebles, J. (2017). Unsupervised visual-linguistic reference resolution in instructional videos. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2183–2192.
- Jermurawong, J. and Habash, N. (2014). Predicting the structure of cooking recipes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2370–2377.
- Kiddon, C., Ponnuraj, G. T., Zettlemoyer, L., and Choi, Y. (2015). Mise en place: unsupervised interpretation of instructional recipes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 982–992.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations*.
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollar, P. (2014). Microsoft coco: common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755.
- Mishra, B. D., Huang, L., Tandon, N., Yih, W. T., and Clark, P. (2018). Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1595–1604.
- Mori, S., Maeta, H., Yamakata, Y., and Sasada, T. (2014). Flow graph corpus from recipe texts. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 2370–2377.
- Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the Conference of Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533.
- Nishimura, T., Hashimoto, A., and Mori, S. (2019). Procedural text generation from a photo sequence. In *Proceedings of the International Conference on Natural Language Generation*.
- Papadopoulos, D. P., Uijlings, J. R., Keller, F., and Ferrari, V. (2017). Extreme clicking for efficient object annotation. In *Proceedings of the International Conference on Computer Vision*, pages 4930–4939.
- Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the International Conference on Computer Vision*, pages 2641–2649.
- Real, E., Shlens, J., Mazzocchi, S., Pan, X., and Vanhoucke, V. (2017). Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 5296–5305.
- Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.
- Su, H., Deng, J., and Fei-Fei, L. (2012). Crowdsourcing annotations for visual object detection. In *Proceedings of the Workshops at the AAAI Conference on Artificial Intelligence*.
- Tandon, N., Mishra, B. D., Grus, J., Yih, W. T., Bosselut, A., and Clark, P. (2018). Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 57–66.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, pages 2048–2057.
- Yagcioglu, S., Erdem, A., Erdem, E., and Ikişler-Cinbis, N. (2018). Recipeqa: A challenge dataset for multimodal

- comprehension of cooking recipes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368.
- Zhang, Y. and Lu, H. (2018). Deep cross-modal projection learning for image-text matching. In *Proceedings of the European Conference on Computer Vision*, pages 686–701.