A Weak supervision with Syntactic Cues for Reference Resolution

Anonymous submission

Abstract

In recipes, contextual understanding of instructions depends on temporal interpretation of the entities because of their spatio-temporal changes. Accordingly, we propose the use of reference resolution to find the origin action of entities, provided that the entity is an output from a previous action, instead of being a raw ingredient. Here, we introduce a weak supervision method that exploits syntactic features for producing latent links between entities and their origin actions. The results show that our weak supervision outperforms the previous unsupervised studies with %8 F1. In particular, our approach indicates %82 resolution performance on pronoun, and %85 on null entities.

1 Introduction

Many studies have been using the captions of the videos to obtain joint embeddings spaces (Miech et al., 2019; Sun et al., 2019; Miech et al., 2020; Zhu and Yang, 2020), or utilizing the descriptive sentences of the instructions for object grounding (Zhou et al., 2018a; Sadhu et al., 2020). Besides, multimodal inputs are used in many language tasks such as video question answering (Zeng et al., 2017; Le et al., 2020), machine translation (Sigurdsson et al., 2020; Gu et al., 2021), and so on. Recipe videos provide rich visual and language data, however one particular challenge is required to be considered: resolving the references of entities.

Linguistic ambiguities (e.g., “the cubes" in Figure 1(c)) are presented in cooking instructions of videos since the spatio-temporal changes of the entities are inevitable. The choose of referring expressions might differ with respect to the changes of the entities 1. As shown with Figure 1, (a) the same nominal phrase refers to a different object (the whole salmon piece; and then one of the halves) whereas in (b) a coreferential pronoun is used although the object has changed (c) is in fact the most well-behaved in terms of keeping the language expressions consistent across actions and with the entities being referred to.

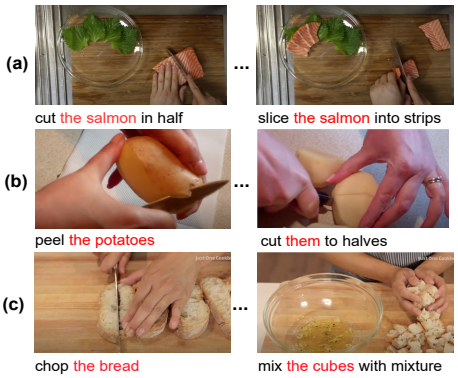

Figure 1: Examples from YouCookII dataset to show the effects of temporal changes on the entities and the referring expressions. Three rows display there different use of expressions of entities.

There has been a few attempts (Kiddon et al., 2015; Huang et al., 2017) to address the reference resolution with unsupervised graph optimization problem in order to find the most likely edges between entity and action nodes of recipes. Kiddon et al. (2015) apply the conditional probability with the given predicate-entity pairs of steps, and entity-action pairs of possible references. Additionally, Huang et al. (2017) adapt the likelihood functions from (Kiddon et al., 2015) and make use of the visual inputs of given actions. As an alternative to graph optimization, Huang et al. (2018) propose an entity-action pointer network to find the origin.

We argue with the above methods since the order of the instructions, utilized ingredients of the same dishes differ according to personal preference. Thus, the assumption of obtaining the same graph for the same dishes breaks the performance of optimization of the action graph. However, we leverage syntactic cues of the instructions for annotating the references for weak supervision.

2 Problem Statement

2.1 Problem Definition

Each recipe consists of ordered instruction steps, where each step s, e.g. *pour olive oil on the Italian bread cubes and bake them in the oven*, includes N number of actions, e.g. two actions occur together in one step like *pour olive oil on the Italian bread cubes* and *bake them in the oven*. Accordingly, each step s of given recipe is segmented into actions a and each action a_i defined as the pair of predicate p_i and the undergoing entity e_i. For example, the first action of the fourth step on Figure 2 e_i denotes the *the onion rings* and p_i refer the verb *move*.

$$\mathbf{s} = a_1, ..., a_N,\ 0 < N,\ \mathbf{a}_j = (p_j, e_j)$$

where p specifies the predicate of the action a_j, whereas e defines the corresponding entity. Reference resolution task is formulated as a function α to find a link from the considered entity e_i to origin action a_o that is one of the previous actions and outputs the e_i.

$$a_o = \alpha(e_i, a_1, ..., a_{i-1})$$

The function α of reference resolution links the entity e_i to most likely action a_o, (i.e. $e_i \rightarrow a_o$). Thus, the latent link is defined from the corresponding entity e_i, e.g., *the dressing*, to its origin action, e.g., mix yogurt and vinegar. However, the raw ingredients need to be neglected linking to any actions since the raw ingredients are not produced by any of the actions. For example, the entity *dry bread crumbs* of the third action in Figure 2 is a raw ingredient which is not produced by any previous actions in the recipe.

2.2 Evaluation

We compute the F-score for evaluation of reference resolution as it is denoted in the previous reference resolution studies in recipes (Kiddon et al., 2015; Huang et al., 2017, 2018) where precision P indicates how many of all the resolved references are correct with the formula $P = \frac{tp}{tp+fp}$ whereas recall R measure how many of the all references are correctly resolved with the formula $R = \frac{tp}{tp+fn}$ where tp designates the number of references that are correctly resolved, fp is the number of references that are not reference (e.g. raw ingredients) but recognized as reference, fn is the number of reference that are not detected as reference. The raw ingredients are out of the evaluation, the references are considered to compute the F-score.

1. crack *an egg* into *a bowl* and break *it*
2. pour *dry bread crumbs* into *the bowl*
3. season *the egg* with *salt* and *spices*
4. coat *onion rings* in *batter* and transfer *them*
5. move *the onion rings* and coat evenly
...

Figure 2: An example of steps in a recipe to present the difference between single and consecutive actions.

3 The Syntactic Structure of Steps

The construction of descriptive sentences of instructions, in Figure 2, differs with respect to single or consecutive actions. The sequential order of single actions may change according to personal preference. For example, the sequence of chopping the tomatoes and peeling the potatoes differs even in the same dishes. However, consecutive actions need to occur in the same order even in different recipes. Before stirring the onion in the pan of oil, it needs to be chopped into pieces first. The consecutive actions sequence the actions that are applied to the *the same entity*.

Single actions. Single action applies only one process to an entity in a step and the process continues with an other entity of the recipe. As can be seen with the third step of Figure 2, *pour dry bread crumbs into the bowl* and *season the egg with salt and spices* are a sample of single actions.

Consecutive actions. Consecutive actions include more than one process applying to the same entities in a step, i.e., $N > 1$. In the first step of Figure 2, *crack* is processed on *the egg* and then *break* applied on the same entity. Here, we combine these two actions into the same step because the entity is the same potatoes even though the predicates are different. We call this self-preference of combining the actions on the same entities *referential tendency* of consecutive actions. The use of null entity and pronouns is very common in consecutive actions. The first and fourth steps of Figure 2 shows the use of pronouns whereas the fifth step indicate the referential tendency of null entity with the predicate *coat* and the first action *move the onion rings* in the same step. The common occurrence (i.e., 35% of the captions in train data) of consecutive actions arise a need of use for weak annotation. SpaCy (Honnibal et al., 2020) is used for determining the consecutive actions and segmenting the steps into individual actions.

4 Weak Supervision with Syntactic Cues

The main goal of weakly supervised modelling is to reduce the need of annotated data for supervised training. A particular instance of weak supervision is using the heuristic-based labeling with linguistic features of data for automatic labeling. In order to make use of the linguistic features for training a reference resolution model, we leverage the syntactic structure of the steps for weak supervised training.

Let the binary label for each pair is assigned either REF for positive instances or a label ¬REF for a negative instances depending on whether or not the a_j is origin for e_i. In single actions, there is no syntactic cues to find the origin action of the entities. Thus, all previous actions are needed to be considered positive candidates $P(\text{REF}|\langle e_i, a_j\rangle)$ where $0 < j < i$. To resolve the entity *a kettle of water* in Figure 3, we need to consider the actions *peel the potatoes* and *cut them to halves* are positive candidates or define the entity as a raw ingredient.

On the other hand, consecutive actions in the same steps provide useful referential tendency to annotate the latent temporal links between entities and their origin actions. In Figure 2, the entity *them* in the second action of fourth step is the output of the first action *coat onion rings in batter* or the *null entity* of the second action of the fourth step is the output of the first action *move the onion rings* in the same step. Therefore, we annotate the entities with the references by $P(\text{REF}|\langle e_i, a_{i-1}\rangle)$ as a positive instance and the negative instances $P(\neg\text{REF}|\langle e_i, a_j\rangle)$ where $0 < j < i - 1$.

5 Experimental Setup

5.1 Dataset

The caption annotation of the YouCookII (Zhou et al., 2018b) dataset is used for this work. The data consists of 2000 cooking videos with the annotation of instruction steps. Each video instruction includes 3 to 15 steps, where each step is an imperative sentence and temporally aligned to the corresponding video segment. The evaluation set (Huang et al., 2018) including 90 videos of YouCookII.However, the steps are decomposed in to actions manually during reference annotation. Therefore, we do not observe the step structure in the evaluation dataset. Each entity is linked to the origin by using the number (i.e., id of action) of the origin action, if it is not a raw ingredient.

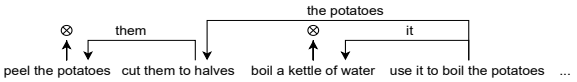

Figure 3: A sample of reference resolution. ⊗ shows the raw ingredients and the links indicate the reference

.

5.2 Input

Train and test instances for the reference resolution are constructed based on entity and the candidate action pairs $\{e_i, a_j\}$. In order to obtain the vector representation (wordEmb(.)) the head of the entity is used for e_i and a_j. Null entities are presented with one time generated random vector.

5.3 Model

Reference resolution is the process in which we identify the origin action that is referring the considered entity e_i. For each candidate a we first encode the actions with $\phi_a(a_j) = [\text{wordEmb}(p_j), \text{wordEmb}(e_j)])$. Each e is represented by $\phi_e(e_i) = \text{wordEmb}(e_i)$.

$$u_{ij} = [w_e \cdot \text{FFNN}_e(\phi_e(e_i)), w_a \cdot \text{FFNN}_a(\phi_a(a_j))]$$

where FFNN denotes a linear feed-forward layer. The input of the model $\phi_e(e_i)$ is the vector representation of i-*th* entity whereas $\phi_a(a_j)$ candidate vector of j-*th* action. w_e is the weights of entity whereas w_a is the weights of action.

$$P(\text{REF}|u_{ij}) = \log(softmax(w \cdot \text{FFNN}(u_{ij})))$$

Thus, the cross-entropy loss are averaged for each batch with the given observations across $\{e_i, a_j\}$ for training the model. To test the model, we start the iteration with e closest candidate a_j and stop the iteration when $P(\text{REF}|e_i, a_j)$ and output $e_i \rightarrow a_j$. If all candidates result ¬REF then the e_i is accepted as a raw ingredient.

5.4 Experiments

Generally speaking, the employed predicates and entities of different recipes are similar. For example, the predicate *chop* might be applied to many different entities, e.g., $a_1 = (\text{chop}, \text{onion})$ and $a_2 = (\text{chop}, \text{greens})$. For an entity example, onion is also used with many different predicates such as chop and stir. Noted similarities arise a key challenge for reference resolution. Therefore, we analyse the use of different word representation such as sub-word, lexical and contextual embeddings. So, we define wordEmb function here.

	100 % annot.	60 % annot.	20 % annot.	w/o annot.	Our Experiments			
Previous.	F1	F1	F1	F1	Exp.	P	R	F1
VLRR	0.56	0.53	0.53	0.51	$\text{RR}_{lexical}$	0.65	0.52	0.58
PNRR	0.59	0.59	0.53	0.49	$\text{RR}_{context}$	0.74	0.47	0.58

Table 1: Results of the reference resolutions. The previous works VLRR and PNRR are presented with different fraction of used labeled data for training. The works are trained by using YouCookII (Zhou et al., 2018b) and tested on the reference annotation dataset (Huang et al., 2018). The results of the previous works are delivered from their own studies. Results of our experiments are produced by the average of three train-test runs.

$\text{RR}_{lexical}$: Reference resolution with lexical features. The input words are represented with the concatenated average embeddings FastText (Bojanowski et al., 2017) and GLoVe (Pennington et al., 2014) to capture sub-word and lexical similarities respectively.

$\text{RR}_{context}$: Reference resolution with contextual features. Base BERT (Devlin et al., 2018) is used to represent the contextual features of the entities whereas FastText used for sub-word representation. In order to encode sub-word with contextual features, we concatenate BERT and FastText of words.

6 Results and Analysis

Results. The aim of this study is to investigate the use of syntactic cues in weak supervision for reference resolution in recipes. Table 1 shows the results of reference resolution for previous studies and our experiments. Visual-linguistic reference resolution (VLRR) (Huang et al., 2017) proposes an unsupervised method by using a joint visual-linguistic features to train expectation-maximization model to optimize the recipe graph. The Pointer network reference resolution (PNRR) (Huang et al., 2017) applies a pointer network (Vinyals et al., 2015) with hierarchical/sequential encoder of the action representation. VLRR and PNRR both use GloVe (Pennington et al., 2014) embedding to represent predicates and entities for inputs. The fraction of labels on the table indicates the fraction of used labeled data. The full size 1.0 includes 60 recipes. Typically, we need to compare our results with the results of a model trained without annotated data (the column w/o label). However, the others are also included in the Table 1 to show effectiveness of our study.

As can be seen Table 1, our lexical ($\text{RR}_{lexical}$) and context ($\text{RR}_{context}$) reference resolution methods outperform the both previous studies with $\%8$ F1 score when the w/o label column considered. Additionally, the use of annotated data with the VLRR and PNRR, also our results of weak supervision show the significant improvement on the performance when we compare with the unsupervised methods. The performance difference of $\text{RR}_{lexical}$ and $\text{RR}_{context}$ can be observed when the precision (P) and recall (R) scores are compared, even though the results of F1 scores are the same.

Analysis. Our methods present significant performance of resolving references of referring expression of pronouns and null entities. $\text{RR}_{lexical}$ gives $\%82$ of all pronouns are resolved correctly, while $\text{RR}_{context}$ indicates $\%97.5$ of all pronouns are linked to correct source action. Moreover, $\%90.9$ of null entities resolved correctly with lexical model, and it is $\%85$ with context model.

Both $\text{RR}_{lexical}$ and $\text{RR}_{context}$ show higher performance for the similar noun phrases are presented in entity and the origin action like the *the bowl* example in Figure 2 when they refer the same entity. However, the entity *the juice* of the action linked to the origin *Add the clam juice to the pan* correctly resolved with the $\text{RR}_{context}$, whereas it is missed by the $\text{RR}_{lexical}$.

On the other hand, different entities with the same noun phrases create a key problem since the lexical and contextual similarities of strong domain bias. For example, *water* used for boiling egg and *water* for noddle are different entities but our method fail to distinguish them and define as a raw ingredient. Additionally, *mixture* entity is constantly resolved as a raw ingredient when the predicate *add* is used to combine the ingredients.

7 Conclusion and Future Work

To conclude, we propose a weak supervision method for reference resolution in recipes and show the way of annotation by leveraging the syntactic cues of instructions for training. Proposed weak supervision method outperforms the previous unsupervised studies. For the future work in recipes we analyze the effect of visual features for resolution.

8 Ethical and Legal Consideration

In this study, there is no concern with identity characteristics, intellectual property, privacy rights, address of possible harms in any section. The claims in this study match results and the results can be expected to generalize in the same experimental setup. Automatic annotation of the data (section 4) to make use of weak supervision method, preparation of the input (section 5.2) of the model (section 5.3) are clearly defined.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Weiqi Gu, Haiyue Song, Chenhui Chu, and Sadao Kurohashi. 2021. Video-guided machine translation with spatial hierarchical attention network. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 87–92.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. 2018. Finding" it": Weakly-supervised reference-aware visual grounding in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5948–5957.

De-An Huang, Joseph J Lim, Li Fei-Fei, and Juan Carlos Niebles. 2017. Unsupervised visual-linguistic reference resolution in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2183–2192.

Chloé Kiddon, Ganesa Thandavam Ponnuraj, Luke Zettlemoyer, and Yejin Choi. 2015. Mise en place: Unsupervised interpretation of instructional recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 982–992.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. 2020. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9972–9981.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Arka Sadhu, Kan Chen, and Ram Nevatia. 2020. Video object grounding using semantic roles in language description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10417–10427.

Gunnar A Sigurdsson, Jean-Baptiste Alayrac, Aida Nematzadeh, Lucas Smaira, Mateusz Malinowski, Joao Carreira, Phil Blunsom, and Andrew Zisserman. 2020. Visual grounding in video for unsupervised word translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10850–10859.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *arXiv preprint arXiv:1506.03134*.

Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. 2017. Leveraging video descriptions to learn video question answering. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Luowei Zhou, Nathan Louis, and Jason J Corso. 2018a. Weakly-supervised video object grounding from text by loss weighting and object interaction. *arXiv preprint arXiv:1805.02834*.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018b. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598.

Linchao Zhu and Yi Yang. 2020. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755.