Unsupervised Visual-Linguistic Reference Resolution in Instructional Videos

De-An Huang[1], Joseph J. Lim[2], Li Fei-Fei[1], and Juan Carlos Niebles[1]
[1]Stanford University, [2]University of Southern California
dahuang@cs.stanford.edu, limjj@usc.edu, {feifeili,jniebles}@cs.stanford.edu

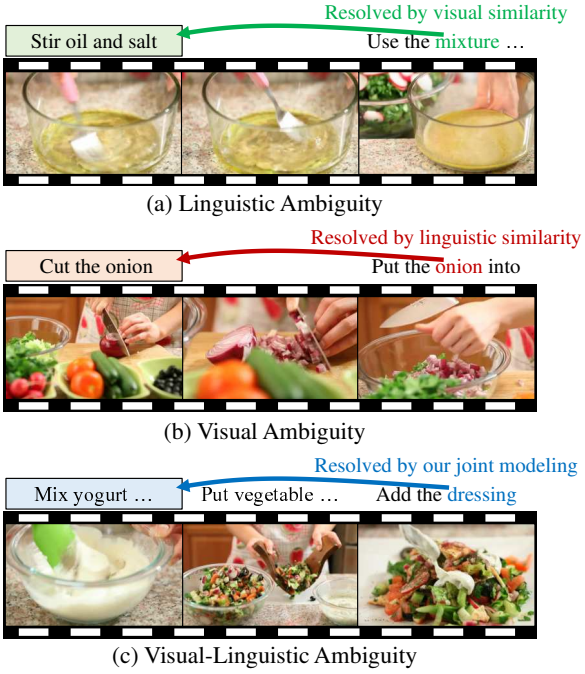
Abstract

We propose an unsupervised method for reference resolution in instructional videos, where the goal is to temporally link an entity (e.g., "dressing") to the action (e.g., "mix yogurt") that produced it. The key challenge is the inevitable visual-linguistic ambiguities arising from the changes in both visual appearance and referring expression of an entity in the video. This challenge is amplified by the fact that we aim to resolve references with no supervision. We address these challenges by learning a joint visual-linguistic model, where linguistic cues can help resolve visual ambiguities and vice versa. We verify our approach by learning our model unsupervisedly using more than two thousand unstructured cooking videos from YouTube, and show that our visual-linguistic model can substantially improve upon state-of-the-art linguistic only model on reference resolution in instructional videos.

1. Introduction

The number of videos uploaded to the web is growing exponentially. In this work, we are particularly interested in the narrated instructional videos. We as humans often acquire various types of knowledge by watching them – from how to hold a knife to cut a tomato, to the recipe of cooking a tomato soup. In order to build a machine with the same capability, it is necessary to understand entities (*e.g.* knife) and actions (*e.g.* cut) in these videos.

From a learning point of view, data from instructional videos pose a very interesting challenge. They are noisy, containing unstructured and misaligned caption uploaded by users or generated automatically by speech recognition. Even worse, the key challenge arises from inevitable ambiguities presented in videos. For example, in Figure 1(a), "oil" mixed with "salt" is later referred as a "mixture" – a linguistic ambiguity due to a referring expression. An onion in Figure 1(b) looks very different from its original appearance before being cut – a visual ambiguity due to a state change. Lastly, "yogurt" is later referred to "dressing" and its appearance changes completely as shown in Figure 1(c) – both linguistic and visual ambiguities.



(a) Linguistic Ambiguity

(b) Visual Ambiguity

(c) Visual-Linguistic Ambiguity

Figure 1. Our goal is to resolve references in videos – temporally linking an entity to the action that produced it. (a), (b), and (c) illustrate challenges resulting from different types of ambiguities in instructional videos and how they are resolved. Our model utilizes linguistic and visual cues to resolve them. An arrow pointing to an action outcome indicates the origin of the entity.

In this paper, we address how to resolve such ambiguities. This task is known as reference resolution: the linking of expressions to contextually given entities [50]. In other words, our goal is to extract all actions and entities from a given video, and resolve references between them. This is equivalent to temporally link each entity (*e.g.* "ice") to the action (*e.g.* "freeze water") that produced it. For example, "mixture" in Figure 1(a) refers to the outcome of the action "stir oil and salt", and "dressing" in Figure 1(c) is the outcome of the action "mix yogurt with black pepper".

There have been various attempts to address reference and coreference resolution in both language understand-

ing [6, 30], and joint vision and language domains [27, 32, 45, 47]. However, most of the previous works either assume that there is enough supervision available at training time or focus on the image-sentence reference resolution, where annotations are easier to obtain. Unfortunately, obtaining high-quality reference resolution annotations in videos is prohibitively expensive and time-consuming.

Thus, in order to avoid requiring explicitly annotated data, we introduce an unsupervised method for reference resolution in instructional videos. Our model jointly learns visual and linguistic models for reference resolution – so that it is more robust to different types of ambiguities. Inspired by recent progress in NLP [23, 39], we formulate our goal of reference resolution as a graph optimization task. In this case, our task of reference resolution is reformulated as finding the best set of edges (i.e. references) between nodes (i.e. actions and entities) given observation from both videos and transcriptions.

We verify our approach using unstructured instructional videos readily available on YouTube [35]. By jointly optimizing on over two thousand YouTube instructional videos with no reference annotation, our joint visual-linguistic model improves 9% on both the precision and recall of reference resolution over the state-of-the-art linguistic-only model [23]. We further show that resolving reference is important to aligning unstructured speech transcriptions to videos, which are usually not perfectly aligned. For a phrase like "Cook it," our visual-linguistic reference model is able to infer the correct meaning of the pronoun "it" and improve the temporal localization of this sentence.

In summary, the main contributions of our work are: (1) introduce the challenging problem of reference resolution in instructional videos. (2) propose an unsupervised graph optimization model using both visual and linguistic cues to resolve the visual and linguistic reference ambiguities. (3) provide a benchmark for the evaluation of reference resolution in instructional videos.

2. Related Work

Coreference/Reference Resolution in Vision In additional to the core task of coreference/reference resolution in NLP [6, 12, 30], there has been recent attempts to address these tasks in conjunction with vision. One task related to our goal of reference resolution in instructional videos is the recent progress on words to image regions reference resolution, where the goal is to spatially localize an object given a referring expression [16, 22, 28, 38, 41, 45, 49, 60, 61]. On the other hand, coreference resolution in texts aligned with the image/video has been shown to be beneficial to the task of human naming [47], image understanding [15], and 3D scene understanding [27]. The most related to our work is the joint optimization of name assignments to tracks and mentions in movies of Ramanathan *et al*. [47]. Nevertheless, our task is more challenging in both the linguistic and visual domains due to the drastic change in both visual appearances and linguistic expression introduced by state changes of the entities.

Instructional Videos. Instructional videos have been used in several contexts in computer vision. The first is semi-supervised and weakly supervised learning, where the transcription is treated as action label without accurate temporal localization [35, 62]. As significant progress has been made on classifying temporally trimmed video clips, recent works aim to obtain the procedural knowledge from the instructional videos [2, 3, 52]. Our goal of reference resolution in instructional videos is a step further as it requires the explicit expression of what action to act on which entities.

Procedural Text Understanding. Our goal of resolving reference in transcription of instructional videos is related to the procedure text understanding in the NLP community [4, 18, 23, 29, 33, 34, 36]. While most approaches require supervised data (ground truth graph annotation) during training [18, 29, 34], Kiddon *et al*. proposed the first unsupervised approach for recipes interpretation [23]. The linguistic part of our approach is inspired by their model. However, as we would show in the experiments, the joint modeling of language and vision plays an important role to interpret the noisier transcription in online videos.

Learning from Textual Supervision. Our learned visual model needs to observe fine-grained details in a frame based on textual supervision to improve reference resolution. This is related to recent progress on aligning and matching textual description with image [19, 54] or video [8, 9, 42, 59, 63]. Another line of work aim to learn visual classifiers based on only textual supervision [5, 7, 11, 48]. Our visual model is trained only with the transcription and is able to help reference resolution in instructional videos.

Extracting Graph from Image/Video. Our formulation of reference resolution as graph optimization is related to the long-standing effort of extracting graphs from image/video. This includes recent progress in scene graphs [13, 20, 51, 64], storylines [1, 14, 17, 31, 53], and action understanding [10, 44, 55]. Our approach of extracting graph associating the entities with action outputs is related to works in robotics where the goal is to transform natural language instructions for the robots to execute [26, 32, 56, 58]. It is important to note that our approach is unsupervised while a large part of the graph extraction approaches require graph annotation at the training stage.

3. Model

Our main goal in this paper is resolving references given an instruction video. Given a video, can we identify all references from entities to actions? For example, "dressing" is referring to the outcome of the action "mix the yogurt and

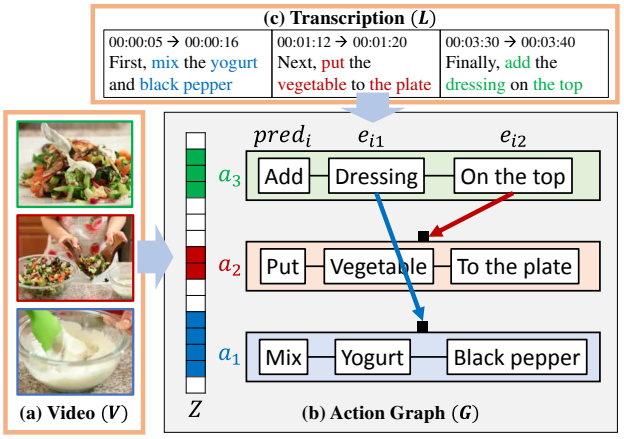


Figure 2. An action graph (G) is a latent representation of references in an instructional video. Both visual (V) and linguistic (L) cues of an instructional video are dependent on an action graph, and they are conditionally independent given an action graph.

black pepper" (shown in Figure 2). Despite its many potential applications, this task comes with two major challenges. First of all, videos contain different types of ambiguities. For example, some entities change their shapes, some are referred by different names, or both. Second, obtaining a large-scale annotation for references in videos is not trivial.

Hence, we propose an unsupervised model for reference resolution. Our model is unique in a way that it (1) learns unsupervisedly, (2) uses both linguistic and visual cues from instructional videos, and (3) utilizes the history of actions to resolve more challenging ambiguities. We formulate our goal of reference resolution as a graph optimization task [39]. More specifically, we use the **action graph** (see Section 3.2) as our latent representation because our goal of reference resolution is connecting entities to action outputs. An overview of our unsupervised graph optimization is shown in Figure 4. We will first describe our model and discuss the details of our optimization in Section 4.

3.1. Model Overview

Our goal is to design an unsupervised model that can jointly learn with visual and linguistic cues of instructional videos. To this end, our model consists of a **visual model** handling video, a **linguistic model** handling transcription, and an **action graph** representation encoding all reference-related information. Our model is illustrated in Figure 2.

In summary, our task is formulated as a graph optimization task – finding the best set of edges (*i.e.* references) between nodes (*i.e.* actions and entities). Essentially, an *action graph* is a latent representation of actions and their references in each video, and observations are made through a video with its visual (*i.e.* frames) and linguistic (*i.e.* instructions) cues; as illustrated in Figure 2. The fact that an *action graph* contains all history information (*i.e.* references over time) helps to resolve a complex ambiguity. Under this formulation, our approach can simply be about learning a likelihood function of an *action graph* given both observations.

Formally, we optimize the following likelihood function:

$$\underset{\mathbf{G}}{\operatorname{argmax}}\, P(\mathbf{L}, \mathbf{V}|\mathbf{G}; \theta_V, \theta_L), \tag{1}$$

where $\mathbf{G}$, $\mathbf{V}$, and $\mathbf{L}$ are the sets of temporally grounded action graph, videos, and corresponding speech transcriptions, respectively. θ_V and θ_L are parameters of visual and linguistic models. Under the assumption that observations are conditionally independent given the action graph, it can be further broken down into

$$\underset{\mathbf{G}}{\operatorname{argmax}}\, P(\mathbf{L}|\mathbf{G}; \theta_L) P(\mathbf{V}|\mathbf{G}; \theta_V). \tag{2}$$

We can thus formulate the visual and linguistic models separately, while they are still connected via an action graph.

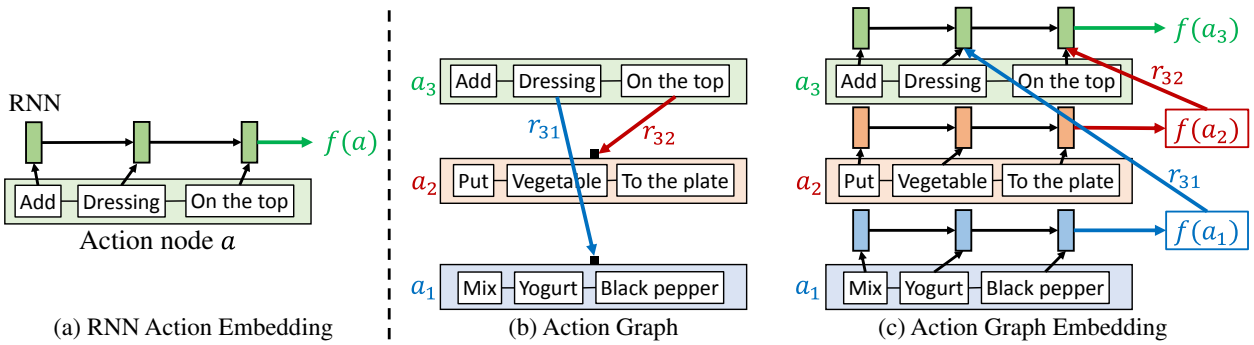
3.2. Temporally Grounded Action Graph (G)

An *action graph* is an internal representation containing all relevant information related to actions, entities, and their references: (1) action description (*e.g.* add, dressing, on the top), (2) action time-stamp, and (3) references of entities. As an example, let's take a look at Figure 2(b), the case of making a salad. Each row represents an action, and each edge from an entity to an action represents a reference to the origin of the entity. Essentially, our goal is to infer these edges (*i.e.* reference resolution). This latent *action graph* representation connects both linguistic and visual models as in Eq. (2). Also, all its reference information later is used to resolve complex ambiguities, which are hard to resolve without the history of actions and references.

To this end, we define *action graph* by borrowing the definition in [23] with a minor modification of adding temporal information. An action graph $G = (E, A, R)$ has $E = \{e_{ij}\}$, a set of **entity nodes** e_{ij}, $A = \{a_i\}$ a set of **action nodes** a_i encompassing and grouping the entity nodes into actions, and $R = \{r_{ij}\}$, a set of edges corresponding to the **references** r_{ij} for each entity e_{ij}. The details are defined as following (See Figure 2(b) for an example):

- $a_i = (pred_i, [e_{ij}], z_i)$: action node
  - $pred_i$: predicate or verb of the action (e.g. put)
  - $e_{ij} = (t_{ij}^{syn}, t_{ij}^{sem}, S_{ij})$: entity nodes of a_i
    - t_{ij}^{syn}: its syntactic type (i.e. $DOBJ$ or PP)
    - t_{ij}^{sem}: its semantic type (i.e. food, location, or other)
    - S_{ij}: its string representation (e.g. [in the bowl])
  - $z_i = (f^{st}, f^{end})$: starting and ending times of a_i
- $r_{ij} = o$: directional edge or reference from entity e_{ij} to its origin action node a_o.

An auxiliary action node a_0 is introduced for entity node not referring to the outcome of another action. For example, if the raw food entity node e_{ij} "chicken" is not coming



(a) RNN Action Embedding (b) Action Graph (c) Action Graph Embedding

Figure 3. (a) We use RNN as the building blocks of our action graph embedding. $f(a)$ is the embedding of action a. (c) shows the action graph embedding of (b). In (c), the embedding of the word "dressing" is averaged with that of its origin, $f(a_1)$, to represent the meaning based on its reference r_{31}. This is then used recursively to compute $f(a_3)$, the embedding of the final step.

from another action, then r_{ij} will connect e_{ij} to a_0. In addition, we allow entity node with empty string representation $S_{ij} = [\phi]$. This can happen when the entity is *implicit* in the transcription. For example, the sentence "Add sugar" implies an implicit entity that we can add the sugar to.

In summary, our action graph is a latent structure that constraints visual and linguistic outputs through $P(L|G;\theta_L)$ and video $P(V|G;\theta_V)$, and also contains all reference information to resolve ambiguities. The definition of *action graph* and its relationships to other models are illustrated in Figure 2. Our goal of reference resolution is reformulated as optimizing the action graph with the highest likelihood given by Eq. (2).

3.3. Visual Model

Visual model $P(V|G;\theta_V)$ is a model that links an action graph to visual cues (*i.e.* video frames). The motivation of our visual model is that it can help resolving linguistic-based ambiguities, and an action graph constrains visual outputs. In other words, our visual model computes a likelihood of an action graph given a set of video frames, where θ_V is the parameters of the model.

For a video $V = [x_1, \ldots, x_T]$, where x_t is the image frame at time t, and its corresponding action graph G, we decompose $P(V|G;\theta_V)$ frame by frame as:

$$P(V|G;\theta_V) = \prod_{t=1}^{T} P(x_t|H_{\bar{z}_t}) \tag{3}$$

where $H_i = (a_{1:i}, r_{1:i})$ is the subgraph before action i, and $\bar{z}_t$ is the action label of frame t. That means $\bar{z}_t = i$ if frame t belongs to action i. $\bar{z}_t = 0$ corresponds to the background.

The key novelty of our visual model is the joint formulation of frame x_t and the corresponding subgraph $H_{\bar{z}_t}$. This formulation is vital to our success of improving reference resolution using visual information. Consider the final action "add dressing on the top" in Figure 2(b). If we swap the references of "dressing" and "on the top", then it will induce a very different meaning and thus visual appearance of this action (i.e. adding vegetable on top of yogurt, instead of adding yogurt on top of vegetable). Our use of $H_{\bar{z}_t}$ instead of $a_{\bar{z}_t}$ in the visual model catches these fine-grained differences and helps reference resolution; setting our approach apart from previous joint image-sentence models.

To compute $P(x_t|H_{\bar{z}_t};\theta_V)$, we learn a joint embedding space for video frames and action (sub)graphs, inspired by visual-semantic embedding works [24, 54]. In other words, we learn θ_V that can minimize the cosine distances between action graph features and visual frame features.

Action Graph Embedding. In order to capture the different meanings of the action conditioned on its references, we propose a recursive definition of our action graph embedding based on RNN-based sentence embedding [25]. Let $g(\cdot)$ be the function of RNN embedding that takes in a list of vectors and output the final hidden state h. Our action graph embedding $f(\cdot)$ is recursively defined as:

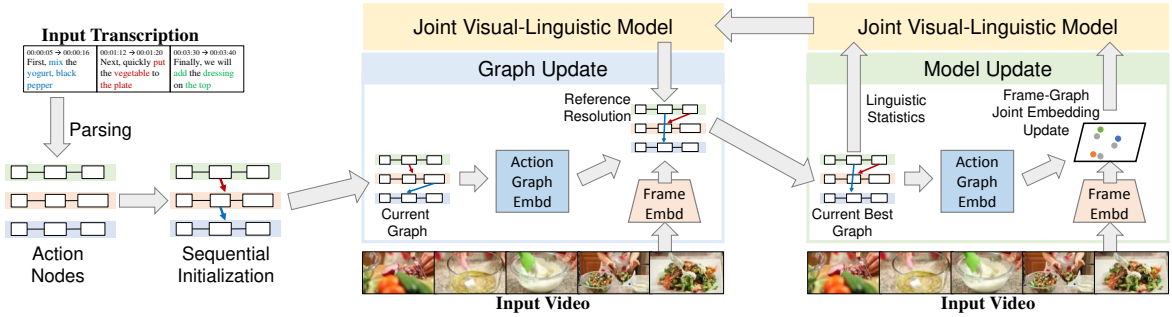
$$f(a_i) = g\left(\left[W(pred_i), \left[W(e_{ij}) + f(a_{r_{ij}})\right]\right]\right), \tag{4}$$

where W is the standard word embedding function [40, 43], and r_{ij} is the origin of e_{ij}. In other words, compared to the standard sentence embedding, where the embedding of e_{ij} is simply $W(e_{ij})$, we enhance it by combining with $f(a_{r_{ij}})$, the embedding of the action it is referring to. This allows our action graph embedding to capture the structure of the graph and represent different meaning of the entity based on its reference. An example is shown in Figure 3.

Frame Embedding We use a frame embedding function from the image captioning models [21, 57]. By transforming the responses of convolutional layers into a vector, it has been shown to capture the fine-grained detail of the image.

3.4. Linguistic Model

Similar to the visual model, our linguistic model $P(L|G;\theta_L)$ links an action graph to linguistic observation. In our case, we use transcripts L of spoken instructions in



(a) Graph Initialization (b) Graph Update (E-step) (c) Model Update (M-step)

Figure 4. An overview of our optimization. (a) We first initialize the graph by just the transcription. We alternate between (b) updating the graph with visual-linguistic reference resolution, and (c) updating the model using visual cues and linguistic statistics in the current graph

videos as our linguistic observation. Then, we know that an action graph will constrain what kind of instructions will be given in the video. Essentially, the linguistic model computes the likelihood of an action graph given transcriptions of the instructional video.

We decompose the linguistic model as follow:

$$P(L|G;\theta_L) = P(L, Z_L|A, R, Z;\theta_L) \propto P(L|A;\theta_L)P(A|R;\theta_L)P(Z_L|Z;\theta_L), \quad (5)$$

where Z_L is the time-stamps of L, and A, R, Z are the actions, references, and time-stamps of the action graph G, respectively. We assume the conditional independence of the time-stamps and that R is independent of L given A.

Here, $P(L|A)$ parses the action nodes from transcriptions using the Stanford CoreNLP package [37].

$P(A|R)$ measures the likelihood of the references given the actions. We adapt the model of Kiddon *et al*. [23] and refer the readers to their paper for details. Briefly, the key models we use are:

- *Verb Signature Model* to capture the property of the verb. For example, "add" tend to combine two food entities.
- *Part-Composite Model* to represent the probable ingredients of an entity. For example, the dressing is more likely to be made up of oil compared to beef.
- *Raw Food Model* to determine if an entity is an action outcome. For example, "flour" is less likely to be an action outcome compared to "dough."

We measure $P(Z_L|Z)$ independently for each action i, where $P(z_{Li}|z_i)$ is defined as:

$$P(z_{Li}|z_i) \propto e^{-\frac{|f^{st}_{Li} - f^{st}_i|}{\sigma}} e^{-\frac{|f^{end}_{Li} - f^{end}_i|}{\sigma}} \quad (6)$$

4. Learning & Inference

We have discussed how we formulate references in instructional videos by the latent structure of an action graph. Using this model, our goal of reference resolution is essentially the optimization for the most likely action graph given the videos and transcriptions based on Eq. (2).

The first challenge of optimizing Eq. (2) is that both the action graph G, and the model parameters θ_L, θ_V are unknown because we aim to learn reference resolution in an unsupervised manner without any action graph annotation.

We thus take a hard EM based approach. Given the current model parameters θ_V and θ_L, we estimate the temporally grounded graphs $\mathbf{G}$ (Section 4.2). Fixing the current graphs $\mathbf{G}$, we update both the visual and linguistic models (Section 4.3). An overview of our optimization is shown in Figure 4. In the following, we will describe our initialization, inference, and learning procedures in more details.

4.1. Graph Initialization

Initially, we have neither an action graph G nor model parameters θ_V and θ_L. Hence, we initialize an action graph G based on a text transcription as the following.

A list of actions A is extracted using Stanford CoreNLP and the string classification model [23]. To simplify our task, we do not update A from the initial iteration. This means all actions we consider are grounded in the transcription. A reference r of each action is initialized to one of the entities in its next action. This is proved to be a strong baseline because of the sequential nature of instructional videos [23]. A temporal location z of each action is initialized as the time-stamp of the action in the transcription.

4.2. Action Graph Optimization (E-step)

In this section, we describe our approach to find the best set of action graphs $\mathbf{G}$ given model parameters θ_V and θ_L. This is equivalent to find the best set of references R and temporal groundings Z for actions in each G, because the set of actions A is fixed from initialization. Jointly optimizing these variables is hard, and hence we relax this to finding the best R and Z alternatively.

Our reference optimization is based on a local search strategy [23]. We exhaustively update the graph with all possible swapping of two references in the current action graph, and update the graph if a reference swapped graph

has a higher probability based on Eq. (2). This process is repeated until there is no possible update.

To optimize our temporal alignment Z, we compute the probabilities of actions for each time based on a language model Eq. (6) and a visual model Eq. (3). Then, we can use dynamic programming to find the optimal assignment of Z to each time based on Eq. (2).

4.3. Model Update (M-step)

Given the action graphs, we are now ready to update our linguistic and visual models.

Linguistic Model Update. We use the statistics of semantic and syntactic types of entities for the verb signature model. For part-composite model, we use Sparse Determinant Metric Learning (SDML)[46] to learn a metric space where the average word embedding of origin's food ingredients is close to that of the current entity e_{ij}. We use logistic regression to classify if the argument is a raw food.

Visual Model Update Given the temporally grounded action graph, for each frame x_t, we are able to get the corresponding subgraph $H_{\bar{z}_t}$. With it as the positive example, we collect the following negative example for our triplet loss: (1) $\tilde{H}_{\bar{z}_t}$, which is the perturbed version of $H_{\bar{z}_t}$. We randomly swap the connections in $H_{\bar{z}_t}$ to generate $\tilde{H}_{\bar{z}_t}$ as negative example. (2) H_i, where $i \neq \bar{z}_t$, subgraph corresponding to other frames are also negative examples. Using the positive and negative examples, we are able to update all our embeddings using backpropagation of triplet loss.

5. Experiments

Given an entity such as "dressing", our goal is to infer its origin – one of the previous actions. We formulate this as a graph optimization problem, where the goal is to recover the most likely references from entities to actions given the observations from transcriptions and videos. We perform the optimization *unsupervisedly* with no reference supervision. In addition to our main task of reference resolution, we show that referencing is beneficial to the alignment between videos and transcriptions.

Dataset. We use the subset of ∼2000 videos with user uploaded caption from the WhatsCookin dataset [35] for our unsupervised learning. Because there is no previous dataset with reference resolution, we annotate reference resolution labels on this subset for evaluation. We use k-means clustering on the captions to select 40 videos, and annotate action nodes A, their temporal locations Z, and references R. This results in 1135 actions, more than two thousand entities and their references. Note that this annotation is just for evaluation, and we do *not* use this annotation for training.

Implementation Details. Our visual embedding is initialized by the image captioning model of [21]. Our linguistic model is initialized by the recipe interpretation model of [23]. All models use learning rate 0.001. For models involving both visual and linguistic parts, we always use equal weights for $P(L|G)$ and $P(V|G)$.

5.1. Evaluating Reference Resolution

Experimental Setup. For evaluation, we first run our model unsupervisedly on all the instructional videos in the dataset. The action and entity nodes here are generated automatically by the Stanford CoreNLP parser [37]. The semantic types of the entities are obtained using unsupervised string classification [23]. After the optimization is finished, we apply one E-step of the final model to the evaluation set. In this case, we use the action and entity nodes provided by the annotations to isolate the errors introduced by the automatic parser and focus on evaluating the reference resolution in the evaluation set. We use the standard precision, recall, and F1 score as evaluation metric [23].

Baselines. We compare to the following models:
- *Sequential Initialization.* This baseline seeks for the nearest preceding action that is compatible for reference resolution, which is a standard heuristic in coreference resolution. This is used as the initial graph for all the other methods.
- *Visual/Linguistic Model Only.* We evaluate in separation the contribution of our visual and linguistic model. Our linguistic model is adapted from [23]. We additionally incorporate word embedding and metric learning to improve its performance in instructional videos.
- *Raw Frame Embedding Similarity (RFES).* We want to know if direct application of frame visual similarity can help reference resolution. In this baseline, the visual model $P(V|G)$ is reformulated as:

$$P(V|G) \propto \prod_{(i,j)\in\mathcal{A}} \prod_{\bar{z}_t=i,\bar{z}_\tau=j} s(x_t, x_\tau), \tag{7}$$

where $s(\cdot,\cdot)$ is the cosine similarity between the frame embeddings given by [21] and $\mathcal{A}$ is the set of all the action pairs that are connected by references in G. In other words, RFES model evaluates the likelihood of a graph by the total visual similarities of frames connected by the references.
- *Frame Embedding Similarity (FES).* We extend RFES to FES by optimizing $s(\cdot,\cdot)$ during the M-step to maximize the probability of the current graphs. In this case, FES is trained to help reference resolution based on frame-to-frame similarity. We compare to this baseline to understand if our model really captures fine-grained details of the image beyond frame to frame visual similarity.
- *Visual+Linguistic w/o Alignment.* Our unsupervised approach faces the challenge of misaligned transcriptions and videos. We evaluate the effect of our update of Z to the reference resolution task.

Results. The results are shown in Table 1. By sequential initialization, we already have a reasonable performance be-

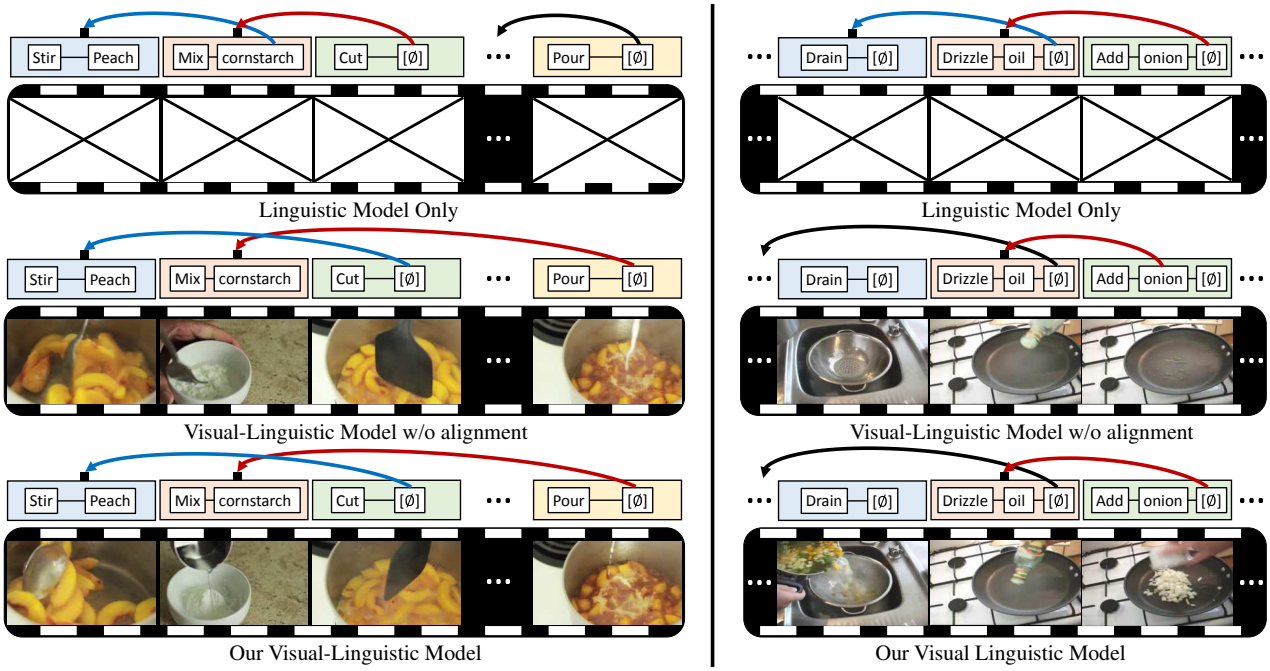


Figure 5. Our reference resolution results. Each row shows the outputs of a type of our model. The first row is of the linguistic only model. For both videos, it fails to resolve long range references. Now, adding the visual information (the 2nd row), our model can resolve longer range references. For example, in the left video, our model can correctly infer the third step is cutting peach (output two steps ahead) using the visual cue. Finally, we show the effect of having alignment in the process of visual-linguistic reference resolution (the 3rd row). For the right video, when the onion appears, our model recognizes that it should be another entity $\emptyset$, rather than onion, that refers "drizzle oil".

Methods	P	R	F1
Sequential Initialization	0.483	0.478	0.480
Random Perturbation	0.399	0.386	0.397
Our Visual Model Only	0.294	0.292	0.293
Our Linguistic Model Only [23]	0.621	0.615	0.618
RFES + Linguistic w/o Align	0.424	0.422	0.423
FES + Linguistic w/o Align	0.547	0.543	0.545
Our Visual + Linguistic w/o Align	0.691	0.686	0.688
Our Visual + Linguistic (Our Full)	**0.710**	**0.704**	**0.707**

Table 1. Reference resolution results. Our final model significantly outperforms the linguistic only model. Note that using vision to help reference resolution is non-trivial. Directly adding frame similarity based visual models is not improving the performance.

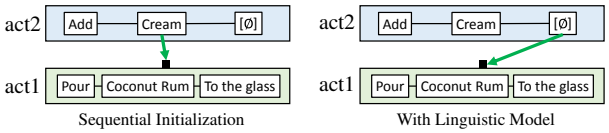


Figure 6. Qualitative results of the linguistic model. $[\phi]$ stands for the implicit entity. On the left, the sequential baseline reference "cream" as the previous action outcome without understanding that it is a raw ingredient. On the other hand, our linguistic model understands (1) cream is raw ingredient, and further (2) "add" is usually used to combine food entities, and thus is able to infer the reference of the implicit entity correctly.

cause of the sequential nature of instruction. This is verified by the fact that if we perform random perturbation to this graph (maximum 10 edge swaps in this case), the reference resolution performance actually goes down significantly. Optimizing using just the visual model for this problem, however, is not effective. Without proper regularization provided by the transcription, the visual model is unable to be trained to make reasonable reference resolution. On the other hand, by using only our linguistic model, the performance improves over sequential baseline by resolving references including common pronoun such as "it", or figuring out some of the words like "flour" is more likely to be raw ingredients and is not referring back to previous action outcomes. Qualitative comparison of the linguistic model is shown in Figure 6.

Importance of our action graph embedding. Direct application of initial frame-level model RFES to the linguistic model, however, cannot improve the reference resolution. This is due to the visual appearance changes caused by the state changes of the entities. The extension of FES improves the performance by 10% compared to RFES since FES optimizes the frame similarity function to help reference resolution. Nevertheless, it is still unable to improve the performance of the linguistic model because whole-frame simi-

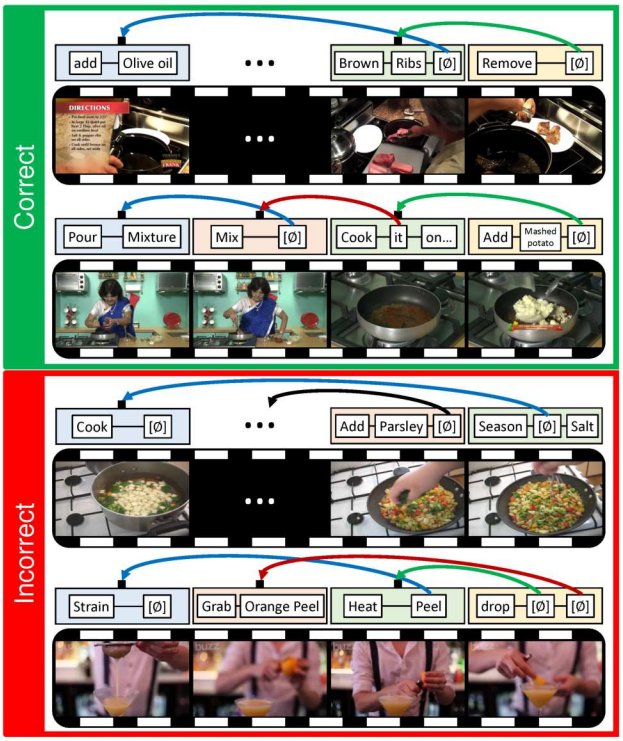


Figure 7. Our reference resolution results. Top two rows show correct references across visual appearances. Bottom two rows show the failure cases of our model. Our visual model can sometimes be confused by similar visual appearances.

larity based model cannot capture fine-grained details of the graph and differentiate references from the same step. On the other hand, our visual model addresses both the challenge introduced by state changes and the fine-grained details of the graph by matching frames to our proposed action graph embedding. In this case, our joint visual-linguistic model further enhances the performance of linguistic model by associating the same entity across varied linguistic expressions and visual appearances that are hard to associate based on only language or frame similarity.

Alignment can help reference resolution. We further verify that the joint optimization with temporal alignment Z can improve the performance of our joint visual-linguistic reference resolution. In this case, as the corresponding frames are more accurate, the supervision to the visual model is less noisy and results in improved performance. Qualitative results are shown in Figure 5 to verify the improvement of both our joint visual-linguistic modeling and video-transcription alignment. Figure 7 shows more qualitative results and failure cases.

5.2. Improving Alignment by Referencing

As discussed earlier, the alignment between captions and frames are not perfect in instructional videos. We have shown that having the alignment in the visual-linguistic model is able to improve reference resolution. In this section, we show that resolving reference is actually also beneficial to improving alignment.

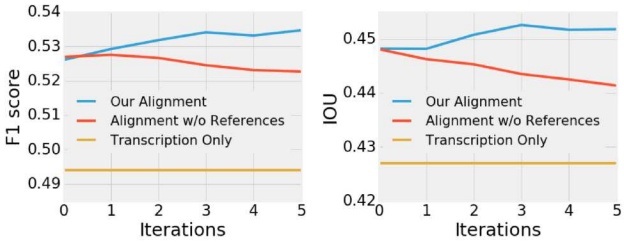


Figure 8. Video to transcription alignment results. By resolving the reference of words in the transcription, our visual-linguistic model is able to improve alignment performance over standard sentence embedding based approach.

Intuitively, for a sentence like "Cut it.", without figuring out the meaning of "it", it is unlikely to train a good visual model because "it" can be referring to food ingredient with a variety of visual appearances. This unique challenge makes the task of aligning transcription with videos more challenging compared to aligning structured text.

To verify our claim, we remove the reference resolution component in our model as the baseline. In this case, the graph embedding is reduced to the standard visual-semantic embedding of [24] without the connections to the previous actions introduced by referencing. For the metrics, we follow previous works [2, 52] and use F1 score and IOU.

The alignment results are shown in Figure 8. It can be seen that without reference resolution in the process of aligning transcription and video, the visual-semantic embedding [24] is not able to improve over iterations. However, our action graph embedding resolves the references in the unstructured instructional text and is thus able to improve the alignment performance. The alignment performance of the transcription is also shown for reference.

6. Conclusion

We propose a new unsupervised learning approach to resolve references between actions and entities in instructional videos. Our model uses a graph representation to jointly utilize linguistic and visual models in order to handle various inherent ambiguities in videos. Our experiments verified that our model can substantially improve upon having only one set of cues to extract meaningful references.

Acknowledgement. This research was sponsored in part by grants from the Stanford AI Lab-Toyota Center for Artificial Intelligence Research, the Office of Naval Research (N00014-15-1-2813), and the ONR MURI (N00014-16-1-2127). We thank Max Wang, Rui Xu, Chuanwei Ruan, and Weixuan Gao for efforts in data collection.

References

[1] H. Agrawal, A. Chandrasekaran, D. Batra, D. Parikh, and M. Bansal. Sort story: Sorting jumbled images and captions into stories. In *EMNLP*, 2016.

[2] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016.

[3] J.-B. Alayrac, J. Sivic, I. Laptev, and S. Lacoste-Julien. Joint discovery of object states and manipulating actions. *arXiv:1702.02738*, 2017.

[4] J. Andreas and D. Klein. Alignment-based compositional semantics for instruction following. In *EMNLP*, 2015.

[5] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *CVPR*, 2004.

[6] A. Björkelund and J. Kuhn. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *ACL*, 2014.

[7] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *ICCV*, 2013.

[8] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015.

[9] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013.

[10] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *CVPR*, 2016.

[11] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

[12] G. Durrett and D. Klein. Easy victories and uphill battles in coreference resolution. In *EMNLP*, 2013.

[13] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In *CVPR*, 2013.

[14] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009.

[15] M. Hodosh, P. Young, C. Rashtchian, and J. Hockenmaier. Cross-caption coreference resolution for automatic image understanding. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 2010.

[16] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. *arXiv preprint arXiv:1611.09978*, 2016.

[17] T.-H. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. Girshick, X. He, P. Kohli, D. Batra, et al. Visual storytelling. In *NAACL*, 2016.

[18] J. Jermsurawong and N. Habash. Predicting the structure of cooking recipes. In *EMNLP*, 2015.

[19] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*. IEEE, 2015.

[21] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[22] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[23] C. Kiddon, G. T. Ponnuraj, L. Zettlemoyer, and Y. Choi. Mise en place: Unsupervised interpretation of instructional recipes. In *EMNLP*, 2015.

[24] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2015.

[25] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[26] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[27] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? text-to-image coreference. In *CVPR*, 2014.

[28] J. Krishnamurthy and T. Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *TACL*, 1:193–206, 2013.

[29] T. A. Lau, C. Drews, and J. Nichols. Interpreting written how-to instructions. In *IJCAI*, 2009.

[30] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, 2011.

[31] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, 2014.

[32] C. Liu, S. Yang, S. Saba-Sadiya, N. Shukla, Y. He, S.-C. Zhu, and J. Y. Chai. Jointly learning grounded task structures from language instruction and visual demonstration. In *EMNLP*, 2016.

[33] R. Long, P. Pasupat, and P. Liang. Simpler context-dependent logical forms via model projections. In *ACL*, 2016.

[34] H. Maeta, T. Sasada, and S. Mori. A framework for procedural text understanding. In *Proceedings of the 14th International Conference on Parsing Technologies*, 2015.

[35] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. Whats cookin? interpreting cooking videos using text, speech and vision. In *NAACL HLT*, 2015.

[36] J. Malmaud, E. J. Wagner, N. Chang, and K. Murphy. Cooking with semantics. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, 2014.

[37] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[38] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

[39] S. Martschat and M. Strube. Latent structures for coreference resolution. *TACL*, 3:405–418, 2015.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[41] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016.

[42] I. Naim, Y. C. Song, Q. Liu, L. Huang, H. Kautz, J. Luo, and D. Gildea. Discriminative unsupervised alignment of natural language instructions with corresponding video segments. *NAACL HLT*, 2015.

[43] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[44] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014.

[45] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

[46] G.-J. Qi, J. Tang, Z.-J. Zha, T.-S. Chua, and H.-J. Zhang. An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization. In *ICML*, 2009.

[47] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking people in videos with their names using coreference resolution. In *ECCV*, 2014.

[48] V. Ramanathan, P. Liang, and L. Fei-Fei. Video event understanding using natural language descriptions. In *ICCV*, 2013.

[49] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016.

[50] D. Schlangen, T. Baumann, and M. Atterer. Incremental reference resolution: The task, metrics for evaluation, and a bayesian filtering model that is sensitive to disfluencies. In *SIGDIAL*, 2009.

[51] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, 2015.

[52] O. Sener, A. R. Zamir, S. Savarese, and A. Saxena. Unsupervised semantic parsing of video collections. In *ICCV*, 2015.

[53] G. A. Sigurdsson, X. Chen, and A. Gupta. Learning visual storylines with skipping recurrent neural networks. In *ECCV*, 2016.

[54] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2:207–218, 2014.

[55] B. Soran, A. Farhadi, and L. Shapiro. Generating notifications for missing actions: Don't forget to turn the lights off! In *ICCV*, 2015.

[56] S. Tellex, P. Thaker, J. Joseph, and N. Roy. Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*, 94(2):151–167, 2014.

[57] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[58] C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu. Robot learning with a spatial, temporal, and causal and-or graph. In *ICRA*, 2016.

[59] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015.

[60] S. Yang, Q. Gao, C. Liu, C. Xiong, S.-C. Zhu, and J. Y. Chai. Grounded semantic role labeling. In *Proceedings of NAACL-HLT*, 2016.

[61] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV*, 2016.

[62] S.-I. Yu, L. Jiang, and A. Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. In *ACM MM*, 2014.

[63] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015.

[64] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013.