Watch-Bot: Unsupervised Learning for Reminding Humans of Forgotten Actions

Chenxia Wu[1], Jiemi Zhang[2], Bart Selman[3], Silvio Savarese[4] and Ashutosh Saxena[5]

***Abstract*— We present a robotic system that watches a human using a Kinect v2 RGB-D sensor, detects what he forgot to do while performing an activity, and if necessary reminds the person using a laser pointer to point out the related object. Our simple setup can be easily deployed on any assistive robot.**

Our approach is based on a learning algorithm trained in a purely unsupervised setting, which does not require any human annotations. This makes our approach scalable and applicable to variant scenarios. Our model learns the action/object co-occurrence and action temporal relations in the activity, and uses the learned rich relationships to infer the forgotten action and the related object. We show that our approach not only improves the unsupervised action segmentation and action cluster assignment performance, but also effectively detects the forgotten actions on a challenging human activity RGB-D video dataset. In robotic experiments, we show that our robot is able to remind people of forgotten actions successfully.

I. Introduction

The average adult forgets three key facts, chores or events every day [1]. Hence it is important for a personal robot to be able to detect not only what a human is currently doing but also what he forgot to do. For example in Fig. 1, someone fetches milk from the fridge, pours the milk to the cup, takes the cup and leaves without putting back the milk, then the milk would go bad. In this paper, we focus on detecting these forgotten actions in the complex human activities for a robot, which learns from a completely unlabeled set of RGB-D videos.

There are a large number of works on vision-based human activity recognition for robots. These works infer the semantic label of the overall activity or localize actions in the complex activity for better human-robot interactions [2], [3], [4], assistive robotics [5], [6]. Given the input RGB/RGB-D videos [7], [8], [9], or 3D human joint motions [10], [11], or from other inertial/location sensors [12], [13], they train the perception model using fully or weekly labeled actions [8], [14], [15], or locations of annotated human/their interactive objects [16], [17]. Recently, there are some other works on anticipating human activities for reactive robotic response [18], [5]. However, to enable a robot to remind people of forgotten things, it is challenging to directly use these approaches especially in a completely unsupervised setting.

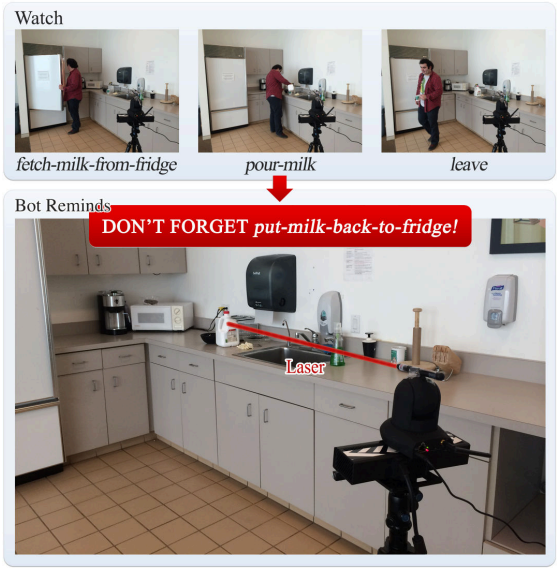


Fig. 1: Our Watch-Bot watches what a human is currently doing, and uses our unsupervised learning model to detect the human's forgotten actions. Once a forgotten action detected (*put-milk-back-to-fridge* in the example), it points out the related object (*milk* in the example) by the laser spot in the current scene.

Our goal is to enable a robot, that we call Watch-Bot, to detect humans' forgotten actions as well as localize the related object in the current scene. The robot consists of a Kinect v2 sensor, a pan/tilt camera (which we call camera for brevity in this paper) mounted with a laser pointer, and a laptop (see Fig. 2(a)). This setup can be easily deployed on any assistive robot. Taking the example in Fig. 1, if our robot sees a person fetch a milk from the fridge, pour the milk, and leave without putting the milk back to the fridge, it would first detect the forgotten action and the related object (the milk), given the input RGB-D frames and human skeletons from the Kinect; then map the object from the Kinect's view to the camera's view; finally pan/tilt the camera until its mounted laser pointer pointing to the milk.

In real robotic applications, people perform a very wide variety of actions. These are hard to learn from existing videos on the Internet and there are few with annotations of actions or objects. So we propose a probabilistic learning model in a completely unsupervised setting, which can learn actions and relations directly from the data without any annotations, only given the input RGB-D frames with tracked skeletons from Kinect v2 sensor.

We model an activity video as a sequence of actions,

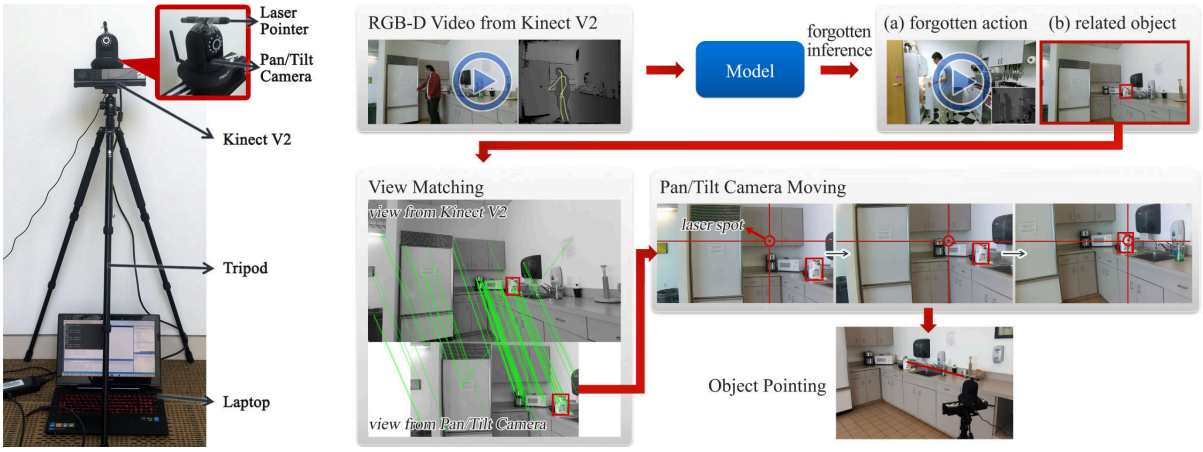
[1]Chenxia Wu is with the Department of Computer Science, Cornell University and Stanford University.{chenxiawu}@cs.cornell.edu

[2]Jiemi Zhang is with Didi Research.{jmzhang10}@gmail.com

[3]Bart Selman is with the Department of Computer Science, Cornell University.{selman}@cs.cornell.edu

[4]Silvio Savarese is with the Department of Computer Science, Stanford University.{ssilvio}@stanford.edu

[5]Ashutosh Saxena is with Brain of Things Inc. {ashutosh}@brainoft.com



(a) Robot System. (b) System Pipeline.

Fig. 2: (a). Our Watch-Bot system. It consists of a Kinect v2 sensor that inputs RGB-D frames of human actions, a laptop that infers the forgotten action and the related object, a pan/tilt camera that localizes the object, mounted with a fixed laser pointer that points out the object. (b). The system pipeline. The robot first uses the learned model to infer the forgotten action and the related object based on the Kinect's input. Then it maps the view from the Kinect to the pan/tilt camera so that the bounding box of the object is mapped in the camera's view. Finally, the camera pan/tilt until the laser spot lies in the bounding box of the target object.

so that we can understand which actions have been taken, *e.g.*, the example activity contains four actions: *fetch-milk-from-fridge*, *pour*, *put-milk-back-to-fridge*, and *leave*.[1] For detecting the forgotten action and reminding, we model the co-occurrence between actions and the interactive objects, as well as the temporal relations between these segmented actions, *e.g.*, action *fetch-milk-from-fridge* often co-occurs with and is temporally after action *put-milk-back-to-fridge*, and object *milk* occurs in both actions. Using the learned actions and relations, we infer the forgotten actions and localize the related objects, *e.g.*, *put-milk-back-to-fridge* might be forgotten as previously seen *fetch-milk-from-fridge* before *pouring*, and seen *leaving* indicates he really forgot to do, also *milk* is the object interacted in the forgotten action.

We evaluate our approach extensively on a large RGB-D human activity dataset recorded by Kinect v2 [19]. The dataset contains 458 videos of human daily activities as compositions of multiple actions interacted with different objects, in which people forgot actions in 222 videos. We show that our approach not only improves the action segmentation and action cluster assignment performance, but also obtains promising results of forgotten action detection. Moreover, we show that our Watch-Bot is able to remind humans of forgotten actions in the real-world robotic experiments.

II. Related Work

Most previous works focus on recognizing human actions for both robotics [2], [8], [9] and computer vision [20], [21], [22]. They model different types of information, such as the temporal relations between actions [23], [24], the human and the interactive object appearances and relations [25], [24]. Yang *et al*. [6] presented a system that learns manipulation action plans for robot from unconstrained youtube videos. Hu *et al*. [15] proposed an activity recognition system trained from soft labeled data for the assistant robot. Chrungoo *et al*. [4] introduced a human-like stylized gestures for better human-robot interaction. Piyathilaka *et al*. [11] used 3D skeleton features and trained dynamic bayesian networks for domestic service robots. However, it is challenging to directly use these approaches for inferring the forgotten actions.

Recently, there are works on anticipating human activities and they performed well for assistant robots [18], [5]. They modeled the object affordances and object/human trajectories to discriminate different actions in past activities and anticipate future actions. However, in order to detect forgotten actions, we also need to consider actions after it such as *boiling water* indicates *filling kettle* before it.

The output laser spot on object is also related to the work 'a clickable world' [26], which selects the appropriate behavior to execute for an assistive object-fetching robot using the 3D location of the click by the laser pointer. Differently, we keep the laser pointer fixed on top of the camera, and pan/tilt the camera iteratively to point out the target object using a real-time view matching.

Most of these works rely on supervised learning given fully labeled actions, or weakly supervised action labels, or locations of human/their interactive objects. Differently, our robot uses a completely unsupervised learning setting that trains model only on Kinect's output RGB-D videos. Our model is based on our previous work [19], which presents a Casual Topic Model to model action relations in the complex activity. In this paper, we further introduce the human interactive object and its relations to actions, so that the robot can localize the related object. We then design a robotic system using the model to kindly remind people.

III. Watch-Bot System

We outline our Watch-Bot system in this section (see Fig. 2). Our goal is to detect what people forgot to do given the observation of his poses and interacted objects. The robot consists of a Kinect v2 sensor, a pan/tilt camera mounted with a laser pointer, and a laptop. The input to our

[1] In the training, we do not know these action semantic labels. Instead we assign the action cluster index.

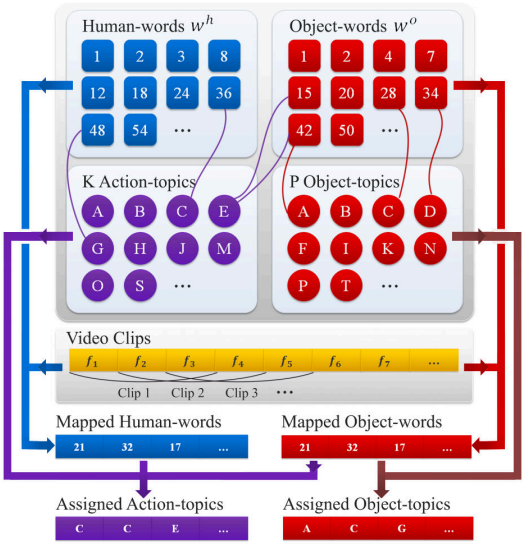


Fig. 3: Video representation in our approach. A video is first decomposed into a sequence of overlapping fixed-length temporal clips. The human-skeleton-trajectories/interactive-object-trajectories from all the clips are clustered to form the human-dictionary/object-dictionary. Then the video is represented as a sequence of human-word and object-word indices by mapping its human-skeleton-trajectories/interactive-object-trajectories to the nearest human-words/object-words in the dictionary. Also, an activity video is about a set of action-topics/object-topics indicating which actions are present and which object types are interacted.

system is RGB-D human activity videos with the tracked 3D joints of human skeletons from Kinect v2. Then we use an unsupervised trained learning model (see Section IV) to infer the forgotten action and localize the related object in the Kinect's view. After that, we map the object bounding box from the Kinect's view to the camera's view. Finally, we pan/tilt the camera until the laser spot lies within the target object in its view (see Section V).

Video Representation. To detect the action structure in the complex activity video, we propose a video representation that draws parallels to document modeling in the natural language [27] (illustrated in Fig. 3). We first decompose a video into a sequence of overlapping fixed-length temporal clips. We then extract the human-skeleton-trajectory features and the interactive-object-trajectory features from the clips. In order to build a compact representation of the activity video, we represent it as a sequence of words. We use k-means to cluster the human-skeleton-trajectories/interactive-object-trajectories from all the clips to form a *human-dictionary* and an *object-dictionary*, where we use the cluster centers as *human-words* and *object-words*. Then, the video can be represented as a sequence of human-word and object-word indices by mapping its human-skeleton-trajectories/interactive-object-trajectories to the nearest human-words/object-words in the dictionary. Also, an activity video is about a set of *action-topics* indicating which actions are present in the video, and a set of *object-topics* indicating which object types are interacted.

Visual Features. We extract both human-skeleton-trajectory features and the interactive-object-trajectory features from the output by the Kinect v2. The new Kinect v2 has high resolution of RGB-D frames (RGB: 1920×1080, depth: 512×424) and improved body tracking of 25 body joints of human skeletons.

We first extract the human-skeleton-trajectory features of the clip as in [19]. Then we extract the human interactive-object-trajectory based on the human hands, image segmentation, motion detection and tracking. We collect the bounding boxes enclosing the potential interested objects from super-pixels output by a fast edge detection approach [28] on both RGB and depth images. We apply the moving foreground mask [29] to remove the unnecessary steady backgrounds and select those segments within a distance to the human hand joints in both 3D points and 2D pixels.

We then track the bounding box in the clip using SIFT matching and RANSAC to get the trajectories. We use the closest trajectory to the human hands for the clip. Finally, we extract six kernel descriptors from the bounding box of each frame in the trajectory: gradient, color, local binary pattern, depth gradient, spin, surface normals, and KPCA/self-similarity, which have been proven to be useful features for RGB-D data [30]. We concatenate the object features of each frame as the interactive-object-trajectory feature of the clip.

IV. Learning Model

We present a new unsupervised model for our Watch-Bot. The graphic model is illustrated in Fig. 4 and the notations are in Table I. Our model is able to infer the probability of forgotten actions using the rich relationships between actions and objects.

We learn the model from a training set of D unlabeled videos. Each video as a document d consists of N_d continuous clips $\{c_{nd}\}_{n=1}^{N_d}$, each of which consists of a human-word w_{nd}^h mapped to the human-dictionary and an object-word w_{nd}^o mapped to the object-dictionary. We assign action-topic to each clip c_{nd} from K latent action-topics, indicating which action-topic they belong to. We assign object-topic to each object-word w_{nd}^o from P latent object-topics, indicating which object-topic is interacted within the clip. The assignments are denoted as $z_{nd}^{(1)}$ and $z_{nd}^{(2)}$. We use superscripts $(1), (2)$ to denote action-topics and object-topics respectively. After assignments, in a video, continuous clips with the same action-topic compose an action segment. All the segments assigned with the same action-topic from the training set compose an action cluster.

As shown in Fig. 4, the generative process of our model is as follows. In a document d, we choose $z_{dn}^{(1)} \sim Mult(\pi_{:d}^{(1)}), z_{dn}^{(2)} \sim Mult(\pi_{:d}^{(2)})$, where $Mult(\pi)$ is a multinomial distribution with parameter π. The human-word w_{nd}^h is drawn from an action-topic specific multinomial distribution $\phi_{z_{nd}^{(1)}}^{(1)}$, $w_{dn}^h \sim Mult(\phi_{z_{dn}^{(1)}}^{(1)})$, where $\phi_k^{(1)} \sim Dir(\beta^{(1)})$ is the human-word distribution of action-topic k, sampled from a Dirichlet prior with the hyperparameter $\beta^{(1)}$. While the object-word w_{nd}^o is drawn from an action-topic and object-topic specific multinomial distribution $\phi_{z_{nd}^{(1)} z_{nd}^{(2)}}^{(12)}$, $w_{dn}^o \sim$

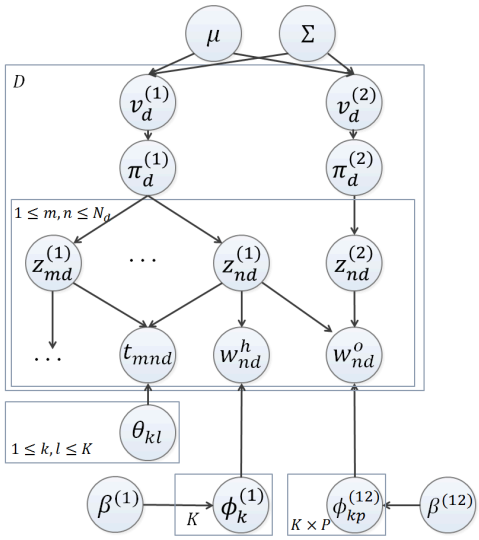


Fig. 4: The probabilistic graphic model of our approach.

$Mult(\phi^{(12)}_{z^{(1)}_{nd} z^{(2)}_{nd}})$, where $\phi^{(12)}_{kp} \sim Dir(\beta^{(12)})$ is the object-word distribution of action-topic k and object-topic p. Here we consider the same object type like *book* can be variant in appearance in different actions such as a *close book* in *fetch-book* and a *open book* in *reading*. So we consider the object-word distribution for different combinations of the action topic and the object topic.

The co-occurrence such as action *put-down-items* and action *take-items*, object *book* and action *reading*, is useful to recognizing the co-occurring actions/objects and gives a strong evidence for detecting forgotten actions. We model the co-occurrence by drawing their priors from a mixture distribution. In the graphic model, $\pi^{(1)}_{kd}, \pi^{(2)}_{pd}$ decide the probability of action-topic k and object-topic p occurring in a document d, where $\sum_{k=1}^{K} \pi^{(1)}_{kd} = 1, \sum_{p=1}^{P} \pi^{(2)}_{pd} = 1$. We construct the probabilities using a stick-breaking process as in [19], where $v^{(1)}_{kd}, v^{(2)}_{pd}$ serve as the priors. Then we draw the packed vector $v_{:d} = [v^{(1)}_{:d}, v^{(2)}_{:d}]$ from a multivariate normal distribution $N(\mu, \Sigma)$, which captures the correlations between action-topics and object-topics.

The temporal relations between actions are also useful to discriminating the actions using temporal ordering and inferring the temporal consistent forgotten actions. So we model the relative time of occurring actions as in [19]. In detail, let $t_{nd}, t_{md} \in (0, 1)$ be the absolute time stamp of n-th clip and m-th clip, which is normalized by the video length. $t_{mnd} = t_{md} - t_{nd}$ is the relative time of m-th clip relative to n-th clip. Then t_{mnd} is drawn from a certain distribution, $t_{mnd} \sim \Omega(\theta_{z^{(1)}_{md}, z^{(1)}_{nd}})$, where $\theta_{z^{(1)}_{md}, z^{(1)}_{nd}}$ are the parameters. $\Omega(\theta_{k,l})$ are K^2 pairwise action-topic specific relative time distributions defined by a product of a Bernoulli distribution which gives the probability of action k after/before the action l, and a normal distribution which estimates how long the action k is after/before the action l.

A. Learning and Inference

We use Gibbs sampling [31], [32] to learn the parameters and the infer the hidden variables from the posterior distribution of our model. The word w^h_{nd}, w^o_{nd} and the relative time t_{mnd} are observed in each video. We can integrate out $\Phi^{(1)}_k, \Phi^{(12)}_{kp}$ since $Dir(\beta^{(1)}), Dir(\beta^{(12)})$ are conjugate priors for the multinomial distributions $\Phi^{(1)}_k, \Phi^{(12)}_{kp}$. We also estimate the standard distributions including the mutivariate normal distribution $N(\mu, \Sigma)$ and the time distribution $\Omega(\theta_{kl})$ using the method of moments, once per iteration of Gibbs sampling. The topic priors $v^{(1)}_{:d}, v^{(2)}_{:d}$ can be sampled by a Metropolis-Hastings independence sampler [33] as in [19]. Following the convention, we use the fixed symmetric Dirichlet distributions by setting $\beta^{(1)}, \beta^{(12)}$ as 0.01.

Then we introduce how we sample the topic assignment $z^{(1)}_{nd}, z^{(2)}_{nd}$. We do a collapsed sampling as in Latent Dirichlet Allocation (LDA) [27] by calculating the posterior distribution of $z^{(1)}_{nd}, z^{(2)}_{nd}$:

$$
\begin{aligned}
&p(z^{(1)}_{nd} = k | \pi^{(1)}_{:d}, z^{(1)}_{-nd}, z^{(2)}_{nd}, t_{nd}) \\
&\quad \propto \pi^{(1)}_{kd} \omega(k, w^h_{nd}) \omega(k, z^{(2)}_{nd}, w^o_{nd}) p(t_{nd} | z^{(1)}_{:d}, \theta), \\
&p(z^{(2)}_{nd} = p | \pi^{(2)}_{:d}, z^{(2)}_{-nd}, z^{(1)}_{nd}) \propto \pi^{(2)}_{pd} \omega(z^{(1)}_{nd}, p, w^o_{nd}), \\
&\omega(k, w^h_{nd}) = \frac{N^{-nd}_{kw^h} + \beta^{(1)}}{N^{-nd}_{k} + N_w \beta^{(1)}}, \\
&\omega(k, p, w^o_{nd}) = \frac{N^{-nd}_{kpw^o} + \beta^{(12)}}{N^{-nd}_{kp} + N_o \beta^{(12)}}, \\
&p(t_{nd} | z^{(1)}_{:d}, \theta) = \prod_{m}^{N_d} \Omega(t_{mnd} | \theta_{z^{(1)}_{md}, k}) \Omega(t_{nmd} | \theta_{k, z^{(1)}_{md}}), \quad (1)
\end{aligned}
$$

where N_w, N_o is the number of unique word types in dictionary, $N^{-nd}_{kw^h}/N^{-nd}_{kpw^o}$ denotes the number of instances of word w^h_{nd}/w^o_{nd} assigned with action-topic k/action-topic k and object-topic p, excluding n-th word in d-th document, and N^{-nd}_{k}/N^{-nd}_{kp} denotes the number of total words assigned with action-topic k/action-topic k and object-topic p. $z^{(1)}_{-nd}/z^{(2)}_{-nd}$ denotes the topic assignments for all words except $z^{(1)}_{nd}/z^{(2)}_{nd}$.

In Eq. (1), note that the topic assignments are decided by which actions/objects are more likely to co-occur in the video (the occurrence probabilities $\pi^{(1)}_{kd}/\pi^{(2)}_{kd}$), the visual appearance of the word (the word distributions

TABLE I: Notations in our model.

Symbols	Meaning
D	number of videos in the training database;
K	number of action-topics;
P	number of object-topics;
N_d	number of human-words/object-words in a video;
c_{nd}	n-th clip in d-th video;
w^h_{nd}	n-th human-word in d-th video;
w^o_{nd}	n-th object-word in d-th video;
$z^{(1)}_{nd}$	action-topic assignment of c_{nd};
$z^{(2)}_{nd}$	object-topic assignment of w^o_{nd};
t_{nd}	normalized timestamp of of c_{nd};
t_{mnd}	$= t_{md} - t_{nd}$ the relative time between c_{md} and c_{nd};
$\pi^{(1)}_{:d}, \pi^{(2)}_{:d}$	the probabilities of action/object-topics in d-th document;
$v^{(1)}_{:d}, v^{(2)}_{:d}$	the priors of $\pi^{(1)}_{:d}, \pi^{(2)}_{:d}$ in d-th document;
$\phi^{(1)}_k$	multinomial human-word distribution from action-topic k;
$\phi^{(12)}_{kp}$	multinomial object-word distribution from action-topic k and object-topic p;
μ, Σ	multivariate normal distribution of $v_{:d} = [v^{(1)}_{:d}, v^{(2)}_{:d}]$;
θ_{kl}	relative time distribution of t_{mnd}, between action-topic k, l;

Algorithm 1 Forgotten Action and Object Detection.

Input: RGB-D video q with tracked human skeletons.
Output: Claim no action forgotten, or output an action segment with the forgotten action and a bounding box of the related object in the current scene.
1. Assign the action-topics to clips and the object-topics to object-words in q as introduced in Section IV-A.
2. Get the action segments by merging the continuous clips with the same assigned action-topic.
3. If the assigned action-topics K_e in q contains all modeled action-topics $[1:K]$, claim no action forgotten and return;
4. For each action segmentation point t_s, each not assigned action-topic $k_m \in [1:K] - K_e$, and each object-topic $p_m \in [1:P]$:
   Compute the probability defined in Eq. 2;
5. Select the top tree possible tuples (k_m, p_m, t_s), and get the forgotten action segment candidate set Q which contains segments with topics (k_m, p_m);
6. Select the top forgotten action segment p from Q with the maximum $forget\_score(p)$;
7. If $forget\_score(p)$ is smaller than a threshold, claim no action forgotten and return;
8. Segment the current frame to super-pixels using edge detection [28] as in Section III;
9. Select the nearest super-pixels to both extracted object bounding box in q and p.
10. Merge the adjacent super-pixels and bound the largest one with a rectangle as the output bounding box.
11. Return the top forgotten action segment and the object bounding box.

$\omega(k, w^h_{nd}), \omega(k, p, w^o_{nd}))$ and the temporal relations (the relative time distributions $p(t_{nd}|z^{(1)}_{:d}, \theta)$). The time complexity of the sampling per iteration is $O(N_d D(\max(N_d K, P))$.

For inference of a test video, we sample the unknown topic assignments $z^{(1)}_{nd}, z^{(2)}_{nd}$ and the topic priors $v^{(1)}_{:d}, v^{(2)}_{:d}$ using the learned parameters in the training stage.

V. Forgotten Action Detection and Reminding

In this section, we describe how we apply our model in our robot to detecting the forgotten actions and reminding people. It is more challenging than conventional action recognition, since what to infer is not shown in the query video. Therefore, unlike the existing models on action relations learning, our model learns rich relations rather than the only local temporal transitions. As a result, those actions occurred with a relatively large time interval, occurred after the forgotten actions, as well as the interactive objects can also be used to detect forgotten actions, *e.g.*, a *put-back-book* might be forgotten as previously seen a *fetch-book* action before a long *reading*, and seen a *book* and a *leaving* action indicates he really forgot it.

Our goal is to detect the forgotten action and then point out the related object in the forgotten action using our learned model (see Alg. 1). We first use our model to segment the query video into action segments (step 1,2 in Alg. 1), and then infer the most possible forgotten action-topic and the related object-topic (step 4 in Alg. 1). Next we retrieve a top forgotten action segment from the training database, containing the inferred forgotten action-topic and the object-topic (step 5,6 in Alg. 1). Using the extracted object in the retrieved segment, we detect the bounding box of the related

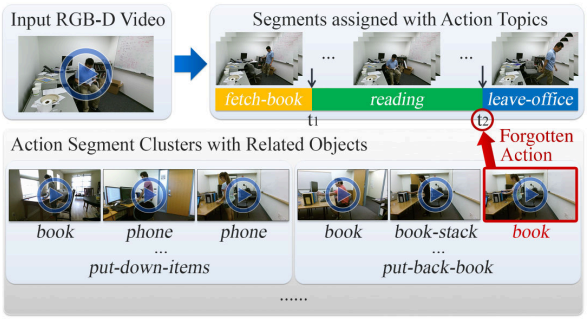


Fig. 5: **Illustration of forgotten action and object detection using our model.** Given a query video, we infer the forgotten action-topic and object-topic in each segmentation point (t_1, t_2). Then we select the top segment from the inferred action-topic's segment cluster with the inferred object-topic with the maximum $forget\_score$.

forgotten object in the Kinect's view of the query video (step 8,9,10 in Alg. 1). After that, we map the bounding box of the object from the Kinect's view to the camera's view. Finally, we pan/tilt camera until its laser pointer points out the related object in the current scene.

Forgotten Action and Object Inference. We first introduce how we infer the forgotten action-topic and object-topic using the dependencies in our learned model. After assigning the action-topics and object-topics to the query video q, we consider adding one additional clip $\hat{c}$ consisting of $\hat{w^h}, \hat{w^o}$ into q in every action segmentation point t_s (see Fig 5). Then the probabilities of the missing action-topics k_m with object-topics p_m in each segmentation point t_s can be computed following the posterior distribution in Eq. (1):

$$
\begin{aligned}
&p(z^{(1)}_{\hat{c}} = k_m, z^{(2)}_{\hat{c}} = p_m, t_{\hat{c}} = t_s | other) \\
&\propto \pi^{(1)}_{k_m d} \pi^{(2)}_{p_m d} p(t_s | z^{(1)}_{:d}, \theta) \sum_{w^h, w^o} \omega(k_m, w^h) \omega(k_m, p_m, w^o), \\
&s.t. \quad t_s \in T_s, \ k_m \in [1:K] - K_e, \qquad (2)
\end{aligned}
$$

where T_s is the set of segmentation points (such as t_1, t_2 in Fig. 5) and K_e is the set of existing action-topics in the video (*fetch-book*, *etc.* in Fig. 5). Thus $[1:K] - K_e$ are the missing topics in the video (*put-down-items*, *etc.* in Fig. 5). $p(t_s|z^{(1)}_{:d}, \theta), \omega(k_m, w^h), \omega(k_m, p_m, w^o)$ can be computed as in Eq. (1). Here we marginalized $\hat{w^h}, \hat{w^o}$ to avoid the effect of a specific human-word or object-word.

Note that, in Eq. (2), the closer topics would have higher probabilities $\pi^{(1)}_{kd}, \pi^{(2)}_{pd}$ to co-occur in this query video as they are drawn from the learned joint distribution. The action-topics which are more consistent with the learned temporal relations would have higher probability $p(t_s|z^{(1)}_{:d}, \theta)$. The marginalized word-topic distribution $\sum_{w^h, w^o} \omega(k_m, w^h) \omega(k_m, p_m, w^o)$ give the likelihood of the topic learned from training data.

Forgotten Action and Object Detection. We then introduce how we retrieve a top action segment from the training database. We first select the top three tuples (k_m, p_m, t_s) using the above probability. These action segments consist a forgotten action candidate segment set Q. We then retrieve the segment from Q with the maximum $forget\_score(p) = ave(\mathbb{D}(f_{pm}, f_{qf}), \mathbb{D}(f_{pm}, f_{qt})) -$

$max(\mathbb{D}(f_{pf}, f_{qt}), \mathbb{D}(f_{pt}, f_{qf}))$, where $\mathbb{D}(,)$ is the average pairwise distances between frames, $ave(,), max(,)$ are the average and max value. The front and the tail of the forgotten action segment f_{pf}, f_{pt} need to be similar to the tail of the adjacent segment in q before t_s and the front of the adjacent segment in q after t_s: f_{qt}, f_{qf}. The middle of the forgotten action segment f_{pm} need to be different to f_{qt}, f_{qf}, as it is a different action forgotten in the video[2]. If the maximum score is below a threshold or there is no missing topics (*i.e.*, $K_e = [1 : K]$) in the query video, we claim there is no forgotten actions.

Then we detect the bounding box of the related forgotten object in the current scene. We segment the current frame into super-pixels as in Section III, then search the nearest super-pixels using the extracted object in the top retrieved action, finally merge the adjacent super-pixels and bound the largest one with a bounding box.

Real Object Pointing. We describe how we pan/tilt the camera to point out the real object. We first compute the transformation homography matrix between the frame of the Kinect and the frame of the pan/tilt camera using keypoints matching and RANSAC, which can be done very fast within 0.1 second. Then we can transform the detected bounding box from the Kinect's view to the pan/tilt camera's view. Since we fix the position of the laser spot in the pan/tilt camera view, next we only need to pan/tilt the camera till the laser spot lies within the bounding box of the target object. To avoid the coordinating error caused by distortion and inconsistency of the camera movement, we use an iterative search plus small step movement instead of one step movement to localize the object (illustrated in Fig. 2). In each iteration, the camera pan/tilt a small step towards to the target object according to the relative position between the laser spot and the bounding box. Then the homography matrix is recomputed in the new camera view, so that the bounding box is mapped in the new view. Until the laser spot is close enough to the center of the bounding box, the camera stops moving.

VI. Experiments

A. Dataset

We evaluate our Watch-Bot in a challenging human activity RGB-D dataset [19] consisting of 458 videos of about 230 minutes in total recorded by the Kinect v2 sensor. Each video in the dataset contains 2-7 actions interacted with different objects (see examples in Fig. 6). We asked 7 subjects to perform human daily activities in 8 offices and 5 kitchens with complex backgrounds and recorded the activities in different views. It is composed of fully annotated 21 types of actions (10 in the office, 11 in the kitchen) interacted with 23 types of objects. The participants finish tasks with different combinations of actions and ordering. Some actions occur together often such as *fill-kettle* and *boil-water*, while some are not always together. Some actions are in a fix order such as *turn-on-monitor* and *turn-off-monitor* while some occur in random order. Also, in the dataset, people forgot actions in 222 videos. There are 3 types of forgotten actions in 'office' and 5 types in 'kitchen'.

[2]Here the middle, front, tail frames are 20%-length of segment centering on the middle frame, starting from the first frame, and ending at the last frame in the segment respectively.

B. Baselines

We compare four unsupervised approaches. They are Hidden Markov Model (HMM) [34], LDA topic model [27], our previous work Causal Topic Model(CaTM) [19] and our Watch-Bot Topic Model (WBTM). We use the same human skeleton and RGB-D features introduced in Section III. In LDA, actions and objects are modeled independently as the priors of action/object assignments are sampled from a fix Dirichlet prior and there is no relative time between actions modeled. For HMM, similarly we set action states which generates both human and object trajectory features of each clip, and object states which generates object trajectory features. Since there is no object modeled in CaTM, we only evaluate its activity related performance.

In the experiments, we set the number of action-topics/object-topics and states for HMM equal to or more than ground-truth action/object classes. For LDA, CaTM and our WBTM, the clip length is set to 20 frames, densely sampled by step one and the size of human/object dictionary is set to 500. The forgotten action candidate set for different approaches consists of the segments with the inferred missing topics by transition probabilities for HMM, the topic priors for LDA. After inference, we use the same forgotten action and object detection method as introduced in Section V.

C. Evaluation Metrics

We test in two environments 'office' and 'kitchen'. In each environment, the dataset is split into a train set with mostly full videos (office: 87, kitchen 119) and a few forgotten videos (office: 10, kitchen 10), and a test set with a few full videos (office: 10, kitchen 20) and mostly forgotten videos (office: 89, kitchen 113). We train the models in the train set and evaluate the following metrics in the test set.

Action Segmentation and Cluster Assignment. As in evaluation for unsupervised clustering, we map the action cluster in the train set to the ground-truth action labels by counting the mapped frames between action-topics and ground-truth action classes as in [19] . Then we can use the mapped action class label for evaluation.

We measure the performance in two ways. Per frame: we compute *frame-wise accuracy (Frame-Acc)*, the ratio of correctly labeled frames. Segmentation: we consider a true positive if the union/intersection of the detected and the ground-truth segments is greater than 40% as in [23]. We compute *segmentation accuracy (Seg-Acc)*, the ratio of the ground-truth segments that are correctly detected and *segmentation average precision (Seg-AP)* by sorting all action segments using the average probability of their words' topic assignments. All above three metrics are computed by taking the average of each action class.

Forgotten Action and Object Detection. We measure the *forgotten action detection accuracy (FA-Acc)* by the portion

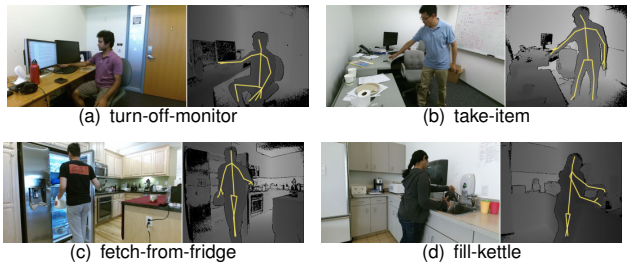


Fig. 6: Action examples in the dataset. The left is RGB frame and the right is depth frame with human skeleton (yellow).

TABLE II: Action segmentation and cluster assignment results, and forgotten action/object detection results.

'office'(%)	Seg-Acc	Seg-AP	Frame-Acc	FA-Acc	FO-Acc
HMM	19.4	23.1	27.3	32.2	20.4
LDA	12.2	19.6	18.4	15.7	10.5
CaTM	32.9	34.6	38.5	41.5	-
WBTM	**35.2**	**36.0**	**41.2**	**46.2**	**36.4**
'kitchen'(%)	Seg-Acc	Seg-AP	Frame-Acc	FA-Acc	FO-Acc
HMM	17.2	18.8	20.3	12.4	5.3
LDA	6.7	17.1	14.4	10.8	5.3
CaTM	29.0	25.5	34.0	20.5	-
WBTM	**30.7**	**28.5**	**36.9**	**24.4**	**20.6**

of correct detected forgotten action or correctly claiming no forgotten actions. We consider the output forgotten action segments by the compared approaches containing over 50% ground-truth forgotten actions as correct. We measure the *forgotten object detection accuracy (FO-Acc)* by the typical object detection metric, that considers a true positive if the overlap rate (union/intersection) between the detected and the ground-truth object bounding box is greater than 40%.

D. Results

Table II, Fig. 7 and Fig. 8 show the main results of our experiments. We discuss our results in the light of the following questions.

How well did forgotten action/object detection perform? In Table II, we can see that our model achieves a promising results for complex activities with multiple objects in variant environments in the completely unsupervised setting. Our models CaTM and WBTM show better performance than traditional uncorrelated topic model LDA, since the co-occurrence and temporal structure are well learned. They outperform HMM, since we consider both the short-range and long-range action relations while HMM only considers the local neighboring states transitions. Our WBTM model improves the performance over CaTM on action clustering and forgotten action detection, also is able to detect the forgotten object, because action and object topics are factorized and their relations are well modeled.

How important is it to consider relations between actions and objects? From the results, we can see that the model which did well in forgotten action detection also performed well in detecting forgotten object. Since our model well considers the relations between the action and the object, it shows better performance in both forgotten action and forgotten object detection than HMM and LDA which

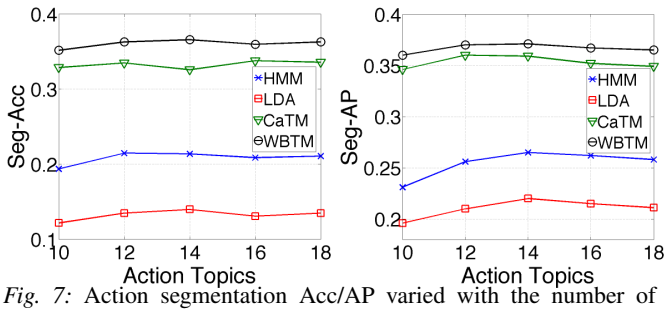


Fig. 7: Action segmentation Acc/AP varied with the number of action-topics in 'office' dataset.

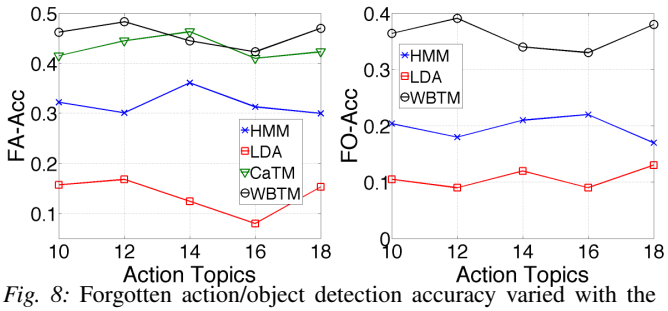


Fig. 8: Forgotten action/object detection accuracy varied with the number of action-topics in 'office' dataset.

models action and object independently as well as CaTM which only models the actions.

How successful was our unsupervised approach in learning meaningful action-topics? From Table II and Fig 7, we can see that the unsupervised learned action-topics can be semantic meaningful even though ground-truth semantic labels are not provided in the training. It can also be seen that, the better action segmentation and cluster assignment performance often indicates better forgotten action detection performance, since actions in the complex activity should be first well segmented and discriminated for next stage forgotten action/object detection.

How did the performance change with the number of action-topics? We plot the performance curves varied with the action-topic number in Fig. 7 and Fig. 8. It shows that the performance does not change much with the action-topics. This is because a certain action might be divided into several action-topics but more variations are also introduced.

E. Robotic Experiments

In this section, we show how our Watch-Bot reminds people of the forgotten actions in the real-world scenarios. We test each two forgotten scenarios in 'office' and 'kitchen' respectively (*put-back-book*, *turn-off-monitor*, *put-milk-back-to-fridge* and *fetch-food-from-microwave*). We use a subset of the dataset to train the model in each activity type separately. In each scenario, we ask 3 subjects to perform the activity twice. Therefore, we test 24 trials in total. We evaluate three aspects. One is objective, the success rate (Succ-Rate): the laser spot lying within the object as correct. The other two are subjective, the average Subjective Accuracy Score (Subj-AccScore): we ask the participant if he thinks the pointed object is correct; and the average Subjective Helpfulness Score (Subj-HelpScore): we ask the participant if the output

Fig. 9: An example of the robotic experiment. The robot detects the human left the food in the microwave, then points to the microwave.

TABLE III: Robotic experiment results. The higher the better.

	Succ-Rate(%)	Subj-AccScore(1-5)	Subj-HelpScore(1-5)
HMM	37.5	2.1	2.3
LDA	29.2	1.8	2.0
WBTM	**62.5**	**3.5**	**3.9**

of the robot is helpful. Both of them are in $1-5$ scale, the higher the better.

Table III gives the results of our robotic experiments. We can see that our robot can achieve over 60% success rate and gives the best performance. In most cases people think our robot is able to help them understand what is forgotten. Fig. 9 gives an example of our experiment, in which our robot observed what a human is currently doing, realized he forgot to fetch food from microwave and then correctly pointed out the microwave in the scene.

VII. Conclusion

In this paper, we enabled a Watch-Robot to automatically detect people's forgotten actions. We showed that our robot is easy to setup and our model can be trained with completely unlabeled videos without any annotations. We modeled an activity video as a sequence of action segments, which we can understand as meaningful actions. We modeled the co-occurrence between actions and the interactive objects as well as the temporal relations between these segmented actions. Using the learned relations, we inferred the forgotten actions and localized the related objects. We showed that our approach improved the unsupervised action segmentation and cluster assignment performance, and was able to detect the forgotten action on a complex human activity RGB-D video dataset. We showed that our robot was able to remind people of forgotten actions in the real-world robotic experiments by pointing out the related object using the laser pointer.

References

[1] "Adults forget three things a day, research finds," http://www.telegraph.co.uk/news/uknews/5891701/Adults-forget-three-things-a-day-research-finds.html, 2009, the Daily Telegraph.

[2] M. Losch, S. Schmidt-Rohr, S. Knoop, S. Vacek, and R. Dillmann, "Feature set selection and optimal classifier for human activity recognition," in *Robot and Human interactive Communication*, 2007.

[3] P. Agarwal, S. Kumar, J. Ryde, J. Corso, and V. Krovi, "Estimating human dynamics on-the-fly using monocular video for pose estimation," in *RSS*, 2012.

[4] A. Chrungoo, S. Manimaran, and B. Ravindran, "Activity recognition for natural human robot interaction," in *Social Robotics*, 2014, vol. 8755, pp. 84–94.

[5] Y. Jiang and A. Saxena, "Modeling high-dimensional humans for activity anticipation using gaussian process latent crfs," in *RSS*, 2014.

[6] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos, "Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web," in *AAAI*, 2015.

[7] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *ICRA*, 2012.

[8] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *IJRR*, vol. 32, no. 8, pp. 951–970, 2013.

[9] G. Chen, M. Giuliani, D. S. Clarke, A. K. Gaschler, and A. Knoll, "Action recognition using ensemble weighted multi-instance learning," in *ICRA*, 2014.

[10] A. Mansur, Y. Makihara, and Y. Yagi, "Action recognition using dynamics features," in *ICRA*, 2011.

[11] L. Piyathilaka and S. Kodagoda, "Human activity recognition for domestic robots," in *Field and Service Robotics*, vol. 105, 2015, pp. 395–408.

[12] L. Chen, J. Hoey, C. Nugent, D. Cook, and Z. Yu, "Sensor-based activity recognition," *SMC*, vol. 42, no. 6, pp. 790–808, 2012.

[13] J.-K. Min and S.-B. Cho, "Activity recognition based on wearable sensors using selection/fusion hybrid ensemble," in *SMC*, 2011, pp. 1319–1324.

[14] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic, "Weakly supervised action labeling in videos under ordering constraints," in *ECCV*, 2014.

[15] N. Hu, Z. Lou, G. Englebienne, and B. Krse, "Learning to recognize human activities from soft labeled data," in *RSS*, 2014.

[16] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *CVPR*, 2013.

[17] B. Ni, V. R. Paramathayalan, and P. Moulin, "Multiple granularity analysis for fine-grained action detection," in *CVPR*, 2014.

[18] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," in *RSS*, 2013.

[19] C. Wu, J. Zhang, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised understanding of actions and relations," in *CVPR*, 2015.

[20] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *ECCV*, 2007.

[21] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *CVPR*, 2012.

[22] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, 2011.

[23] H. Pirsiavash and D. Ramanan, "Parsing videos of actions with segmental grammars," in *CVPR*, 2014.

[24] X. Wang and Q. Ji, "A hierarchical context model for event recognition in surveillance video," in *CVPR*, 2014.

[25] H. S. Koppula and A. Saxena, "Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation," in *ICML*, 2013.

[26] H. Nguyen, A. Jain, C. D. Anderson, and C. C. Kemp, "A clickable world: Behavior selection through pointing and context for mobile manipulation," in *IROS*, 2008.

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[28] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *ICCV*, 2013.

[29] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1999.

[30] C. Wu, I. Lenz, and A. Saxena, "Hierarchical semantic labeling for task-relevant rgb-d perception," in *RSS*, 2014.

[31] D. M. Blei and J. D. Lafferty, "Topic models," *Text mining: classification, clustering, and applications*, vol. 10, p. 71, 2009.

[32] D. I. Kim and E. B. Sudderth, "The doubly correlated nonparametric topic model," in *NIPS*, 2011.

[33] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

[34] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.