Towards a Watson That Sees: Language-Guided Action Recognition for Robots

Ching L. Teo, Yezhou Yang, Hal Daumé III, Cornelia Fermüller and Yiannis Aloimonos

***Abstract*— For robots of the future to interact seamlessly with humans, they must be able to reason about their surroundings and take actions that are appropriate to the situation. Such reasoning is only possible when the robot has knowledge of how the World functions, which must either be learned or hard-coded. In this paper, we propose an approach that exploits language as an important resource of high-level knowledge that a robot can use, akin to IBM's Watson in Jeopardy!. In particular, we show how language can be leveraged to reduce the ambiguity that arises from recognizing actions involving hand-tools from video data. Starting from the premise that tools and actions are intrinsically linked, with one explaining the existence of the other, we trained a language model over a large corpus of English newswire text so that we can extract this relationship directly. This model is then used as a prior to select the best tool and action that explains the video. We formalize the approach in the context of 1) an unsupervised recognition and 2) a supervised classification scenario by an EM formulation for the former and integrating language features for the latter. Results are validated over a new hand-tool action dataset, and comparisons with state of the art STIP features showed significantly improved results when language is used. In addition, we discuss the implications of these results and how it provides a framework for integrating language into vision on other robotic applications.**

I. Introduction

Humans display an uncanny ability to perceive the world that far surpasses current vision only based systems both in terms of precision and accuracy. This is largely due to the vast amount of high-level knowledge that humans have acquired over their lives that enables humans to infer and recognize complex visual inputs. In the same way, service and personal robots of the future must be endowed with such knowledge in order to interact and reason with humans and their environments effectively. This knowledge can be encoded in various forms, of which language is clearly the most predominant. Language manifests itself in the form of text which is also extremely accessible from various large research corpora.

The ability to reason is an essential faculty for service and personal robots. A typical scenario is when the robot needs to understand an action or task which the human performs for the purpose of learning. Such Learning from Demonstration (LfD) [1] paradigm is gaining popularity in robotics as shown by the recently concluded Learning by Demonstration Challenge at AAAI 2011[1]. There are two interconnected components in LfD. The first component recognizes what actions occurred and the second component creates an internal representation so that the action can be recreated by the robot. Unlike the LfD challenge where the robot is supposed to recreate the *one* task it was taught (the second component), we address a variant of the first component where the robot labels the correct action associated with a set of *unlabeled* action data (with repetitions) performed by different human teachers (or actors). Posing our problem in terms of unlabeled data over different actors is also closer to reality as the robot needs to learn how to *generalize* the action so that it can discover, in the second component, an optimal representation to recreate this action on its own.

In this paper, we consider the task of recognizing human activities from videos sequences – specifically, activities that involve hand-tools. Action or activity recognition has remained one of the most difficult problems in computer vision. The main reason is that detections of key objects that define the action in video – tools, objects, hands and humans are still unreliable even using current state of the art object detectors [10], [26]. Descriptions based on tracking trajectories of local features such as STIP [19] and modeling velocity as suggested in [22], are strongly viewpoint dependent and may be confused when similar movements are used for different actions, e.g. drinking from a cup vs. peeling. Both actions involve large up-down hand movements. The main challenge of action recognition is the ambiguity when these two components: objects and trajectories, are considered in isolation. From Fig. 1(top), detecting a cup or action trajectory in isolation can only suggest some likely actions. A more reliable prediction can be achieved when we combine

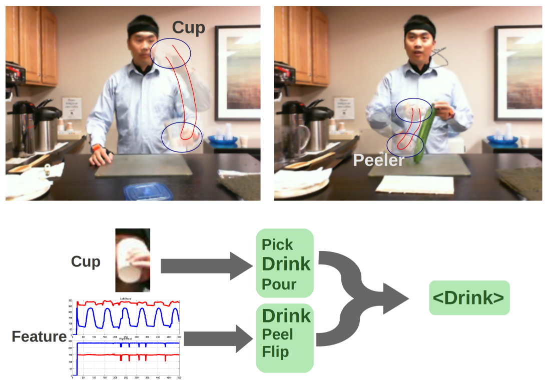


Fig. 1. (Top) Ambiguities in action recognition: similar trajectories for different actions. Tools considered in isolation can only suggest possible actions. (Below) Language can predict, given the tool and action trajectories, the most likely action label.

The authors are from Department of Computer Science, University of Maryland, College Park, MD 20742, USA {cteo,yzyang,hal,fer,yiannis}@umiacs.umd.edu

[1]http://www.lfd-challenge.org/

both object detection and trajectories.

A missing component in the above approach is how we can combine object detection with actions in a logical and intuitive way. To do this, we propose to model language as a resource of prior knowledge that essentially encodes how actions and objects (tools) are related. This language model allows us to weigh the video detections so that the objects and action features that best explain the video are eventually selected (Fig. 1(below)).

II. Related Work

Action recognition research spans a long history. Comprehensive reviews of recent state of the art can be found in [28], [30], [21]. Most of the focus was on studying human actions that were characterized by movement and change of posture, such as walking, running, jumping etc. Many studies represent actions by modeling body poses, body joints and body parts [24]. Depending on the extent of the features used, the literature distinguishes between local and global action models. The former use spatio-temporal interest points and descriptors based on intensity, gradients and flow within spatio-temporal cuboids centered on these interest points [19], [7]. The latter compute descriptors over the whole video frame or an extracted human skeleton or silhouette. For example, [9] used histograms of optical flow and Gorelick et al. [11] and Yilmaz et al. [31] represented human activities by 3-D space-time shapes. Another class of approaches model the evolution of actions over time. For example Bissacco et al. [3] used joint-angle as well as joint trajectories from motion-capture data and features extracted from silhouettes to represent action profiles. Chaudhry et al. [5] employed non linear dynamical systems to model the temporal evolution of optical flow histograms.

Our approach is more closely related to the use of language for object detection and image annotation. With advances on textual processing and detection, several works recently focused on using sources of data readily available "in the wild" to analyze static images. The seminal work of Duygulu et al. [8] showed how nouns can provide constraints that improve image segmentation. Gupta et al. [14] (and references herein) added prepositions to enforce spatial constraints in recognizing objects from segmented images. Berg et al.[2] processed news captions to discover names associated with faces in the images, and Jie et al.[17] extended this work to associate poses detected from images with the verbs in the captions. Some studies also considered dynamic scenes. [6] studied the aligning of screen plays and videos, [20] learned and recognized simple human movement actions in movies, and [15] studied how to automatically label videos using a compositional model based on AND-OR-graphs that was trained on the highly structured domain of baseball videos .

These recent works had shown that exploiting co-occurring text information from scripts and captions aids in the visual labeling task. Our paper takes this further by using *generic* text obtained from the English Gigaword corpus [12], which is a large corpus of English newswire text from which we learn a language model. As we will show, using general NLP tools, we still can derive interesting relationships to guide the visual task of action recognition.

III. Our Approach

The input is a set of $|M|$ videos, $M = \{m_d\}, d \in \{1, 2, \cdots, |M|\}$ containing some actions, with each video containing exactly one unique action. The $|V|$ actions are drawn from the set $V = \{v_j\}, j \in \{1, 2, \cdots, |V|\}$. This means that we have assumed that the task of segmenting actions from a long video sequence has been done. In addition, we assume that every action must have an actor that uses a particular hand-tool. The same tool can be used in multiple actions. The $|N|$ tools comes from the set $N = \{n_i\}, i \in \{1, 2, \cdots, |N|\}$. All labels (both actions and tools) must be known in advance, which is a requirement for learning the appropriate language model. A summary of the approach is shown in Fig. 2.

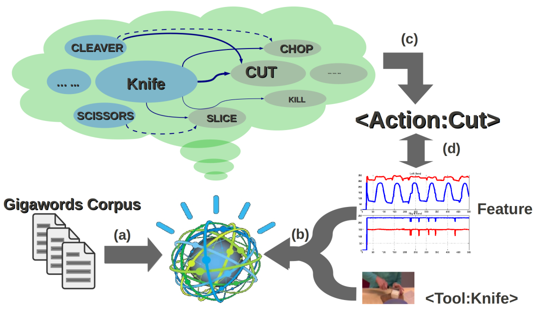


Fig. 2. Key components of the approach.: (a) Training the language model from a large text corpus. (b) Detected tools are queried into the language model. (c) Language model returns prediction of action. (d) Action features are compared and beliefs updated.

The high level overview of the approach is as follows (see Fig. 2): 1) We first detect potential tools from the input video. 2) For each identified tool, we query a trained language model to determine the most likely verbs (actions) associated with the tool. 3) We then confirm the predicted action using the action features obtained from the video to update our confidence on the current action label of the video. This process is repeated until our belief on the action labels is maximized over all tools and action features. Note that our approach is *symmetric*, that is, we could have started off with action features and inquire the language model in exactly the same way. Our choice of starting with objects is based on the fact that object detectors are better researched and are generally more accurate than action detectors.

For the purpose of this discussion, we shall assume the most general case where we only have unlabeled video data. This means that the best that we can do is to perform some form of clustering to *discover* automatically what is the best action label that describes the cluster. Intuitively, we want to learn the "meaning" of actions by grounding them to visual representations obtained from video data. Hence if we knew this grounding, we can assign action labels to the videos. On the other hand, if we know the action labels of the video data, we can estimate this grounding. This leads naturally to an iterative Expectation-Maximization (EM) formulation where we attempt to determine the optimal grounding parameters that will assign action labels to videos with the highest probability.

More formally, our goal is to label each video with their most likely action, along with the tool that is associated with the action. That is, we want to maximize the likelihood:

$$\begin{aligned} L(\mathcal{D}; A) &= \mathbb{E}_{\mathcal{P}(A)}[L(\mathcal{D}|A)] \\ &= \mathbb{E}_{\mathcal{P}(A)}[log\mathcal{P}(F_M, \mathcal{P}_I(\cdot), \mathcal{P}_L(\cdot)|A)] \qquad (1) \end{aligned}$$

where A is the current (binary) action label assignments of the videos (see eq. (3)). $\mathcal{D}$ is the data computed from the video that consists of: 1) the language model $\mathcal{P}_L(\cdot)$ that predicts an action given the detected tool (sec. III-A), 2) the tool detection model $\mathcal{P}_I(\cdot)$ (sec. III-B) and 3) the action features, F_M, associated with the video (sec. III-C). We describe how these 3 data components are computed in the following paragraphs and detail how we optimize eq. (1) using EM in sec. IV-A.

A. *Language as a predictor of actions*

The key component of our approach is the language model that predicts the most likely verb (action) that is associated with a noun (tool) trained from a large text corpus. We view the Gigaword Corpus as a large text resource that contains the information we need to make correct predictions of actions given the detected tools from the video. We do this by training a language model $\mathcal{P}_L(v_j, n_i)$ that returns the maximum likelihood estimates of an action v_j given the tool n_i. This can be done by counting the number of times v_j co-occurs with n_i in a sentence:

$$\mathcal{P}_L(v_j|n_i) = \frac{\#(v_j, n_i)}{\#(n_i)} \qquad (2)$$

As many English words share common meanings, a simple count of the action words (verbs) defined in $\mathcal{V}$ is likely to grossly underestimate the relationship between the tool and the action it is associated with. For example, in the Gigaword Corpus, counting the number of times `drink` co-occurs with `cup` where the actual words are used will not be significantly larger than `pick` and `cup`. The reason is that `cup` can mean a normal drinking cup or a trophy cup. In order to ensure that $\mathcal{P}_L$ captures the correct sense of the word (nouns, verbs), we use WordNet to determine the synonyms and hyponymns of the tools and actions considered. As illustrated in Fig. 3, extending `cup` to include `drinking_glass, wine_glass, mug` captures the expected action `drink` in a larger part of the corpus.

We then recompute $\mathcal{P}_L$ using these enlarged word classes to capture more meaningful relationships between the co-occurring words. Fig. 4 shows the $|N| \times |V|$ co-occurrence matrix of likelihood scores over all the tools and actions considered in the experiments, denoted as $\mathcal{P}_L(V|N)$.

For most of the tool classes, the predicted actions are correct (large values along the diagonals): for e.g. `peeler` predicts `peeling` with high probability (0.94) and `shaker` predicts `sprinkling` at 0.66. This shows that for tools that have a *unique* use, our approach can

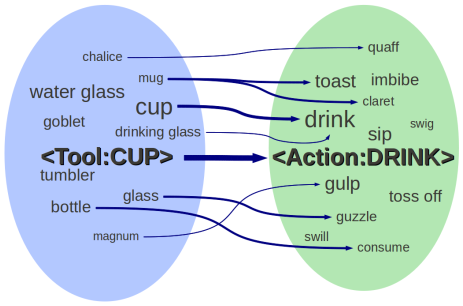


Fig. 3. Enlarging the word class to contain synonyms yields more reasonable counts: `cup` only connects weakly with `drink`. By clustering other closely related words together, their combined counts increase the desired association between `cup` and `drink`.

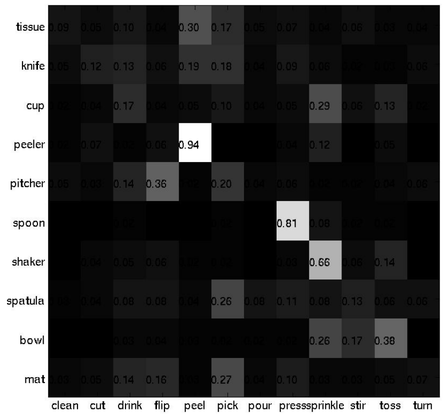


Fig. 4. Gigaword co-occurrence matrix of tools and predicted actions.

predict the expected action easily. However, there are many co-occurrences which we could not anticipate. For e.g, using synonyms of `cup` makes it more selective to `drinking` (0.17) but it is `sprinkling` that has the highest prediction score (0.29). Investigating further reveals that `sprinkling` has some synonyms such as `drizzle moisten splash splosh` which have uses that are also close to `cup`. Other mis-selected tools-action are also due to the confusion at the synonyms/hyponymns levels. We also notice that more *general* actions such as `picking` have a more uniform distribution across the tools, which is expected. In the same way, the tool `mat` is also very general in its use such that it displays no significant selectivity to any of the actions. Despite this simplistic model, most of the entries in $\mathcal{P}_L$ make sense – and it properly reflects the innate complexity of language. As will be shown in sec. V, although the priors from language are weak, they are still helpful for the task of action recognition.

B. *Active tool detection strategy*

We pursue the following active strategy for detecting, and subsequently recognizing, relevant tools in the video as illustrated in Fig. 5. First, a trained person detector [10] is used to determine the location of the human actor in the video frame. The location of the face is also detected using [29]. Optical flow is then computed [4] and we focus on human regions which have the highest flow, indicating the potential locations of the hands. We then apply a variant of a CRF-based color segmentation [23] using a trained skin color+flow model to segment the hand-like regions which are moving. This is justified by the fact that the moving hand is in contact with the tool that we want to identify. In some cases, the face may be detected (since it may be moving) but they are removed using the face detector results. We then apply a trained Partial Least Squares (PLS) object detector similar to [26] near the detected active hand region that returns a detection score at each video frame. Averaging out the detection yields $P_I(n_i|m_d)$, the probability that a tool $n_i \in N$ exists given the video m_d. We denote $P_I(N|M)$ as the set of detection scores (essentially the likelihood) over all tools in N and all videos in M.

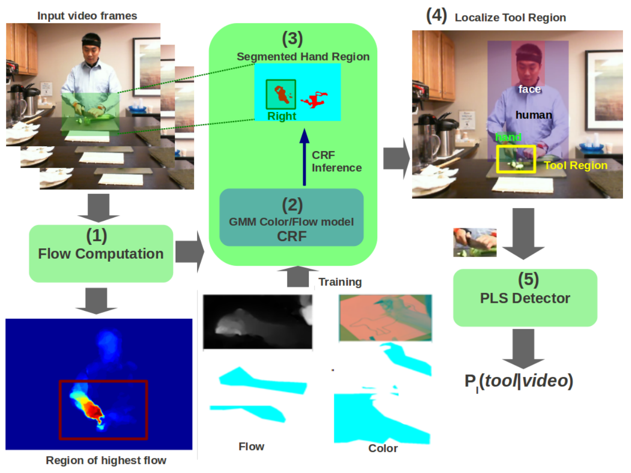


Fig. 5. *(Best viewed in color)* Overview of the tool detection strategy: (1) Optical flow is first computed from the input video frames. (2) We train a CRF segmentation model based on optical flow + skin color. (3) Guided by the flow computations, we segment out hand-like regions (and removed faces if necessary) to obtain the hand regions that are moving (the active hand that is holding the tool). (4) The active hand region is where the tool is localized. Using the PLS detector (5), we compute a detection score for the presence of a tool.

This active approach has two important benefits. By focusing our processing only on the relevant regions of the video frame, we dramatically reduce the chance that the tool detector will misfire. At the same time, by detecting the hand locations, we obtain immediately the action trajectory, which is used to describe the action as shown in the next section.

C. Action features

Tracking the hand regions in the video provides us with two sets of (left and right) hand trajectories as shown in Fig. 6. We then construct for every video a feature vector F_d that encodes the hand trajectories. F_d encodes the frequency and velocity components. Frequency is encoded by using the first 4 real coefficients of the Fourier transform in both the x and y directions, f_x, f_y, which gives a 16-dim vector over both hands. Velocity is encoded by averaging the difference in hand positions between two adjacent frames $\langle \delta x \rangle$, $\langle \delta y \rangle$ which gives a 4-dim vector. These features are then combined to yield a 20-dim vector F_d.

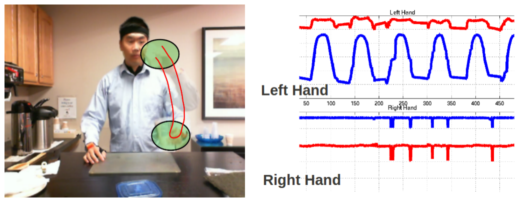


Fig. 6. Detected hand trajectories. x and y coordinates are denoted as red and blue curves respectively.

We denote F_M as the set of of action features F_d over all videos in M.

IV. Using Language to Guide Recognition

In this section, we formalize our EM approach to learn a joint tool-action model that assigns the most likely action label associated with a set of unlabeled video. We first derive from eq. (1) an expression that allows EM to estimate the parameters of this model, followed by details of the Expectation and Maximization steps. We then show how to use the learned model to perform testing. Finally, we consider the case where we have labeled data in which we formulate a simpler supervised approach.

A. Unsupervised learning of a joint tool-action model

We first define the latent assignment variable A. To simplify our notations, we will use subscripts to denote tools $i = n_i$, actions $j = v_j$ and videos $d = m_d$. For each $i \in N$, $j \in V$, $d \in M$, A_{ijd} indicates whether an action j is performed using tool i during video clip d.

$$A_{ijd} = \begin{cases} 1 & j \text{ is performed using } i \text{ during } d \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and A is a 3D indicator matrix (or a 3D array) over all tools, actions and videos. Denoting the parameters of the model as $\mathcal{C} = \{\mathcal{C}_j\}$ which specifies the grounding of each action j, we seek to determine from eq. (1) the maximum likelihood parameter:

$$\mathcal{C}^* = \arg\max_{\mathcal{C}} \sum_A L(\mathcal{D}, A|\mathcal{C}) \tag{4}$$

Where,

$$\begin{aligned} L(\mathcal{D}, A|\mathcal{C}) &= \log P(\mathcal{D}, A|\mathcal{C}) \\ &= \log P(A|\mathcal{D}, \mathcal{C}) P(\mathcal{D}|\mathcal{C}) \end{aligned} \tag{5}$$

with the data $\mathcal{D}$ comprised of the tool detection likelihoods $\mathcal{P}_I(N|M)$, the tool-action likelihoods $\mathcal{P}_L(V|N)$ and action features F_M under the current model parameters $\mathcal{C}$. Geometrically, we can view $\mathcal{C}$ as the superset of the $|V|$ action label centers that defines our current grounding of each action j in the action feature space.

Using these centers, we can write the assignment given *each* video d, tool i and action j, $P(A_{ijd}|\mathcal{D}, \mathcal{C})$ as:

$$\mathcal{P}(A_{ijd} = 1|\mathcal{D}, \mathcal{C}) = \mathcal{P}_I(i|d)\mathcal{P}_L(j|i)Pen(d|j) \tag{6}$$

where $Pen(d|j)$ is an exemplar-based likelihood function defined between the associated action feature of video d, F_d and the current model parameter for action j, $\mathcal{C}_j$ as:

$$Pen(d|j) = \frac{1}{Z} \exp^{-||F_d - \mathcal{C}_j||^2} \tag{7}$$

where Z is a normalization factor. What eq. (7) encodes is the penalty that we score against the assignment when there is a large mismatch between F_d and $\mathcal{C}_j$, the cluster center of action j.

Rewriting eq. (6) over *all* videos M, tools N and actions V we have:

$$\mathcal{P}(A = 1|\mathcal{D}, \mathcal{C}) = \mathcal{P}_I(N|M)\mathcal{P}_L(V|N)Pen(F_M|C) \tag{8}$$

where we use the set variables to represent the full data and assignment model parameters. In the derivation that follows, we will simplify $\mathcal{P}(A = 1|\mathcal{D}, \mathcal{C})$ as $\mathcal{P}(A = 1)$ and $\mathcal{P}(A = 0) = 1 - \mathcal{P}(A = 1)$. We detail the Expectation and Maximization steps in the following sections.

1) Expectation step: We compute the expectation of the latent variable A, denoted by $\mathcal{W}$, according to the probability distribution of A given our current model parameters $\mathcal{C}$ and data ($\mathcal{P}_I$, $\mathcal{P}_L$, and F_M):

$$\begin{aligned} \mathcal{W} &= \mathbb{E}_{\mathcal{P}(A)}[A] \\ &= \mathcal{P}(A = 1) \times 1 + (1 - \mathcal{P}(A = 1)) \times 0 \\ &= \mathcal{P}(A = 1) \end{aligned} \tag{9}$$

According to Eq. 6, the expectation of $\mathcal{A}$ is:

$$\mathcal{W} = \mathcal{P}(A = 1) \propto \mathcal{P}_I(N|M)\mathcal{P}_L(V|N)Pen(F_M|\mathcal{C}) \tag{10}$$

Specifically, for each $i \in N, j \in V, d \in M$:

$$\mathcal{W}_{ijd} \propto \mathcal{P}_I(i)\mathcal{P}_L(j|i)Pen(d|j) \tag{11}$$

Here, $\mathcal{W}$ is a $|N| \times |V| \times |M|$ matrix. Note that the constant of proportionality does not matter because it cancels out in the Maximization step.

2) Maximization step: The maximization step seeks to find the updated parameters $\hat{\mathcal{C}}$ that maximize eq. (5) with respect to $\mathcal{P}(A)$:

$$\hat{\mathcal{C}} = \arg\max_{\mathcal{C}} \mathbb{E}_{\mathcal{P}(A)}[\log \mathcal{P}(A|\mathcal{D}, \mathcal{C})\mathcal{P}(\mathcal{D}|\mathcal{C})] \tag{12}$$

Where $\mathcal{D} = \mathcal{P}_I, \mathcal{P}_L, F_M$. EM replaces $\mathcal{P}(A)$ with its expectation $\mathcal{W}$. As $A, \mathcal{P}_I, \mathcal{P}_L$ are independent of the model parameters $\mathcal{C}$, we can simplify eq. (12) to:

$$\begin{aligned} \hat{\mathcal{C}} &= \arg\max_{\mathcal{C}} \mathcal{P}(F_M|\mathcal{C}) \\ &= \arg\max_{\mathcal{C}} \left(-\sum_{i,j,d} \mathcal{W}_{ijd}||F_d - \mathcal{C}_j||^2 \right) \end{aligned} \tag{13}$$

where we had replaced $\mathcal{P}(F_M|\mathcal{C})$ with eq. (7) since the relationship between F_M and $\mathcal{C}$ is the penalty function $Pen(F_M|\mathcal{C})$. This enables us to define a target maximization function as $\mathbb{F}(\mathcal{C}) = \sum_{i,j,d} \mathcal{W}_{ijd}||F_d - \mathcal{C}_j||^2$.

According to the Karush-Kuhn-Tucker conditions, we can solve the maximization problem by the following constraint:

$$\frac{\partial \mathbb{F}}{\partial \mathcal{C}} = -2 \sum_{i,j,d} (\mathcal{W}_{ijd}(F_d - \mathcal{C}_j)) = 0 \tag{14}$$

Thus, for each $j \in V$, we have:

$$\hat{\mathcal{C}}_j = \frac{\sum_{i \in N, j \in V, d \in M} \mathcal{W}_{ijd} F_d}{\sum_{i \in N, j \in V, d \in M} \mathcal{W}_{ijd}} \tag{15}$$

We then update $\mathcal{C} = \hat{\mathcal{C}}$ within each iteration until convergence.

3) Testing the learned model: The learned model $\mathcal{C}^*$ can be used to classify new videos from a held-out testing set. Denoting the input test video as m_t, we predict the most likely action label, v_t^* by:

$$v_t^* = \arg\max_{j \in V} \sum_{i \in N} \left(\mathcal{P}_I(i|m_t)\mathcal{P}_L(j|i)Pen(F_t|\mathcal{C}_j^*) \right) \tag{16}$$

where F_t is the action features extracted from m_t and $\mathcal{C}_j^*$ is the j action center from the learned model.

B. Supervised action classification

If we have labeled video data of actions, a supervised approach will be the most straightforward. Every video, m_d, is represented by a set of features $\mathcal{F}_d$ that combines F_d (action features), $\mathcal{P}_I$ (tool detection) and $\mathcal{P}_L$ together in the following manner:

$$\begin{aligned} \mathcal{F}_d &= [F_d; \mathcal{P}_I(N|m_d); \mathcal{P}_I(N|m_d) \times \mathcal{P}_L(V|N)] \\ &= [F_d; \mathcal{P}_I(N|m_d); \mathcal{P}_L(V|m_d)] \end{aligned} \tag{17}$$

where ; means a concatenation of the features vectors. This yields a $20 + |N| + |V|$-dim vector. What eq. (17) means is that for every video m_d, we concatenate its F_d together with $\mathcal{P}_I(N|m_d)$, the distribution over all $|N|$ tools, and together with the verb prediction: $\mathcal{P}_L(V|m_d)$, obtained from the text corpus. Given labeled training videos from all possible actions in $\mathcal{V}$, we can proceed to train discriminative classifiers (SVM, Bayes Net and Naive Bayes) to predict the action in the testing video.

V. Experiments

We performed a series of experiments using a new dataset of human actions performed with hand-tools to show quantitatively how language aids in action recognition. We first describe the dataset, and report recognition results on both the unsupervised and supervised scenarios.

A. The UMD Sushi-Making Dataset

The UMD Sushi-Making Dataset[2] consists of 12 actions, performed by 4 actors using 10 different kitchen tools, for the purpose of preparing sushi. This results in 48 video sequences each of around 1000 frames (30 seconds long). Other well known datasets such as the KTH, Weizmann or

[2]http://www.umiacs.umd.edu/research/POETICON/umd_sushi/

Human-EVA datasets [25], [11], [27] do not involve hand-tools. The human-object interaction dataset by Gupta et al. [13] has only 4 objects. The dataset by Messing et al. [22] has only 4 actions with tool use. The CMU Kitchen Dataset [18] has many tool interactions for 18 subjects making 5 recipes, but many of the actions are blocked from view due to the placements of the 4 static cameras. The head mounted camera gives a limited and shaky top-down view which cannot be processed easily.

Our Sushi-Making dataset provides a clear view of the action in use with the tools and it simulates an active robotic agent observing the human actor performing the action in a realistic environment: a kitchen, with a real task: making sushi, that is made up of several actions that the robot must identify. See Fig. 5 for an example. The 12 actions are: cleaning, cutting, drinking, flipping, peeling, picking (up), pouring, pressing, sprinkling, stirring, tossing, turning. The tools used are: tissue, knife, cup, rolling-mat, fruit-peeler, water-pitcher, spoon, shaker, spatula, mixing-bowl. As was discussed in sec. III-A, some of the actions such as `picking` or `flipping` are extremely general and are easily confused. We made this choice to ensure that the language prediction $\mathcal{P}_L$ is *not* perfect and to show that our approach works even under noisy data.

B. Baseline: Vision-only Recognition

As a baseline, we perform experiments without the language component, that is $\mathcal{P}_L$ in eq. (17) is not considered as part of $\mathcal{F}_d$. Two experiments: 1) clustering using K-means and 2) supervised classification are performed.

K-means clustering results: For the case of unlabeled videos, we performed K-means clustering with $k = 12$. We used 36 videos in this experiment. As labels, we took the majority ground truth labels from each cluster to be the predicted labels of the cluster. We then counted the number of videos that are correctly placed in the right cluster to derive a measure of accuracy, which is reported in Fig. 8(a).

Supervised classification results: From the 48 videos from the UMD Sushi-Making dataset, we used 36 videos from 3 actors to train a degree 3 polynomial SVM classifier for the 12 actions. We set the cost parameter to 1.0 with a tolerance termination value at 0.001. These parameters were chosen from a separate development set of 8 videos. The remaining 12 videos were then used for testing. 4-fold cross validation was performed and the classification accuracy is reported. In addition, we trained a Naive Bayes (NB) classifier and a Bayes Net (BN) classifier over the training data. The BN is initialized as a NB with at most 1 parent. We then apply a simple estimator to estimate the conditional probability tables. All classifiers are tested using WEKA [16]. We summarize the results in Fig. 8(b).

C. Adding Language

In this section, we performed experiments with the aid of the language component $\mathcal{P}_L$. Three separate experiments are performed: 1) Unsupervised EM, 2) Semi-Supervised EM where we initialized the model parameters $\mathcal{C}$ with 12 *known* labels and 3) Supervised classification using trained SVM, Bayes Net and Naive Bayes classifiers. 36 videos were used for training the joint tool-action model using EM and 12 videos were held out for testing. For the supervised part, the same parameters as the baseline were used. We report our unsupervised and supervised results in Figs. 8(a) and 8(b) respectively. More detailed results (confusion matrices for each action) can be found online[3]. In addition, we show in Fig. 7 the improving recognition accuracy of the trained model at each iteration. The evolution of the action labels versus the ground truth is also presented. Testing on the held out video set using the trained model yields a recognition accuracy of 83.33%.

D. Comparison with state of art action features

We compared our approach with a bag-of-words (BoW) representation built upon state of the art STIP [19] features clustered using K-means with $k = 50$. We trained three classifiers: SVM, Naive Bayes and Bayesian Net using the same procedure and parameters as the baseline and we summarize the results in Table I. The BoW representation using

Feature	Method	Accuracy
STIP+Bag of Words	Naive Bayes	56.25%
	Bayes Net	75%
	SVM	77.08%
Action Features+Language	Naive Bayes	66.67%
	Bayes Net	85.41%
	SVM	91.67%
Action Features+Language	Unsupervised EM	77.78%
	Semi-supervised EM	91.67%

TABLE I

CLASSIFICATION ACCURACY: STIP VERSUS OUR APPROACH.

STIP achieves a maximum classification rate of 77.08% with trained SVM classifiers. Our approach which uses comparatively *simpler* video features: Fourier coefficients + velocity eq. (17) outperforms the state of the art significantly. This gain is possible due to the addition of language prediction in the action feature.

E. Discussion: the effects of adding language

Comparing the unsupervised recognition results, K-means clustering on the action features alone (with $\mathcal{P}_L$) achieves only 69.44% recognition rate. The clustering accuracy, with the addition of $\mathcal{P}_L$ and using the EM formulation described in sec. IV-A achieves 77.78% with random initialization of the model $\mathcal{C}$. We show further that with correctly initialized parameters from 12 labeled videos, is enough to increase the accuracy to 91.67%, which is just as good as the SVM classifier (which is supervised). This result shows that once again, even with no or limited labeled data, our proposed EM formulation is able to leverage the predictive power of $\mathcal{P}_L$ to find the optimal action and its corresponding tool that best explains the video. Fig. 9 shows some video frames with the predicted action and corresponding tool using EM.

[3]`http://www.umiacs.umd.edu/research/POETICON/umd_sushi/res_ICRA2012`

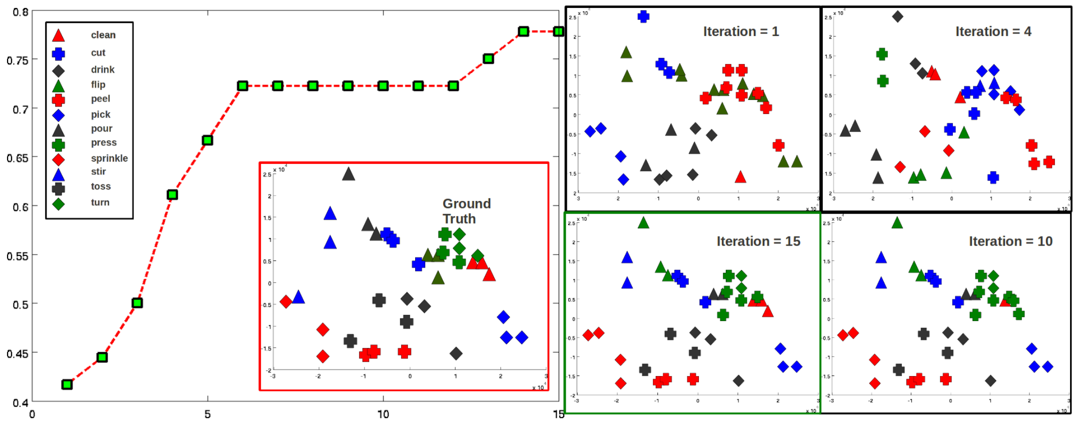


Fig. 7. *(Best viewed in color)* (Left) Unsupervised EM: accuracy at each iteration. (Right) Scatterplots of action label assignments at selected iterations. We see that with each iteration, the assignment label clusters approaches the ground truth label (boxed in red). Note that we used PCA to reduce the action feature dimensions to 2 for visualization.

For the classification results in Fig. 8(b) using the three classifiers, we clearly see that the addition of $\mathcal{P}_L$ increases the classification accuracy, with the most dramatic increase when SVM or Naive Bayes are used: from 87.5% to 91.67% (SVM) and 62.5% to 66.67% when language is added. This shows that even with a simple model, $\mathcal{P}_L$ is able to provide additional discriminatory features which improve the classification. The most important result is that these features are estimated directly from a generic text corpus and the method is not limited to a particular domain such as cooking. This fact alone highlights the strength of language in aiding action classification.

Besides improving action recognition accuracy in both supervised and unsupervised scenarios, another key observation from our results is that language is *complementary* in aiding many vision related tasks where the use of high-level knowledge is required. Previous works described in sec. II have shown that language (in a restricted sense) can be used to simplify ill-posed image problems like segmentation and annotation. We showed here that the difficult problem of recognizing actions involves high-level knowledge as well. This is because of the strong relationship between the actions and the tools that were used to perform these actions. Instead of learning from a huge amount of training *image* data on how tools correlate with actions, we showed that it is possible to obtain such information directly from a *text* corpus. Such a text corpus, although noisy, is much easier to obtain than annotated image data; and we showed that with the right EM formulation, the noisy predictions from $\mathcal{P}_L$ provides us appreciable gains in recognition rates on *unlabeled* video.

VI. Summary and Future Work

This paper has shown a principled approach of integrating large scale language corpora for the purpose of action recognition in videos involving hand-tools. We validated our approach in both supervised and unsupervised scenarios and out-performed the current state-of-the-art STIP+BoW features significantly. These results demonstrate the strength of using language which encodes the intrinsic relationships between tools and actions, leveraging it to aid in the action recognition task. As was advocated in the introduction, our approach has important implications for robots that acquire new skills via LfD, and we are currently evaluating the proposed approach on a mobile robotic agent over an even larger set of actions V and tools N, and developing strategies to detect the tools and actions better across differing viewpoints as the robot moves. We are also investigating the utility of tracking the hand and detecting the tools better by exploiting depth information captured using a Kinect camera.

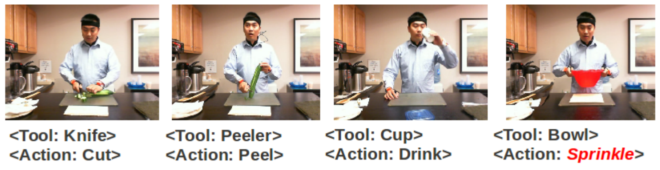
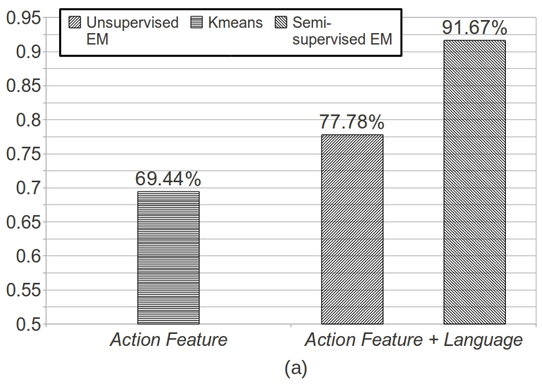


Fig. 9. Some predicted action and tools using EM. The wrong prediction (in red and italicized) of the `sprinkle` action is due to a high co-occurrence with `bowl` in $P_L(V|N)$.

Our approach, however, goes beyond action recognition and can be extended to other vision problems faced in robotics with a more careful treatment of language. Progress in the areas of object recognition, image segmentation and general scene understanding have been slow as these problems require *semantic grounding*. Language, when exploited properly, provides for this. For e.g. using shallow-parsing or Named-Entity Recognition to improve $\mathcal{P}_L$ predictions and subsequently $\mathcal{F}_d$; or performing a dependency parse to reduce the need to use synonyms to extract the tool and related verb from a sentence more accurately. An important limitation of our current approach is that we need to know in advance the action labels and tools of the video. We are currently working on approaches to discover, using *attributes* of the potential tool and action features obtained from the video, a prediction of the tool and action labels directly from language. The potential world-knowledge embedded in lan-



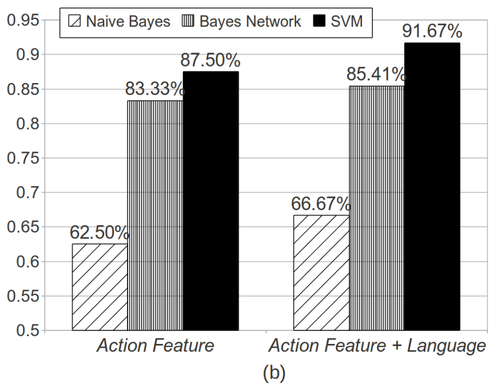


Fig. 8. (a) Unsupervised recognition accuracy: no language (K-Means) versus language (EM). (b) Classification accuracy: no language versus language. All reported results have variances within $\pm 0.5\%$.

guage, along with its complexities, was clearly demonstrated with Watson in Jeopardy! which set a milestone in AI. We believe it will do the same for vision and robotics in the near future.

VII. Acknowledgements

The support of the European Union under the Cognitive Systems program (project POETICON) and the National Science Foundation under the Cyberphysical Systems Program is gratefully acknowledged. Ching Teo and Yezhou Yang are supported in part by the Qualcomm Innovation Fellowship.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 57:469–483, May 2009.

[2] T. L. Berg, A. C. Berg, J. Edwards, and D. A. Forsyth. Who's in the picture? In *NIPS*, 2004.

[3] A. Bissacco, A. Chiuso, and S. Soatto. Classification and recognition of dynamical models: The role of phase, independent components, kernels and optimal transport. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:1958–1972, November 2007.

[4] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *CVPR*, pages 41–48. IEEE, 2009.

[5] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, 2009.

[6] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*. 2008.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

[8] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *ECCV (4)*, volume 2353 of *Lecture Notes in Computer Science*, pages 97–112. Springer, 2002.

[9] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, 2003.

[10] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010.

[11] L. Gorelick, M. Blank, E. Shechtman, R. Basri, and M. Irani. Actions as space-time shapes. *PAMI*, 29(12):2247–2253, 2007.

[12] D. Graff. English gigaword. In *Linguistic Data Consortium, Philadelphia, PA*, 2003.

[13] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*. IEEE Computer Society, 2007.

[14] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In D. A. Forsyth, P. H. S. Torr, and A. Zisserman, editors, *ECCV (1)*, volume 5302 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2008.

[15] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans on PAMI*, 31(10):1775–1789, 2009.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations, Volume 11, Issue 1*, 2009.

[17] L. Jie, B. Caputo, and V. Ferrari. Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In *NIPS*, December 2009.

[18] F. D. la Torre, J. Hodgins, J. Montano, S. Valcarcel, R. Forcada, and J. Macey. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. Technical report, CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, July 2009.

[19] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439. IEEE Computer Society, 2003.

[20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[21] A. P. B. Lopes, E. A. d. Valle, J. M. de Almeida, and A. A. de Arajo. Action recognition in videos: from motion capture labs to the web. (arXiv:1006.3506), Jun 2010. Preprint submitted to CVIU.

[22] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV '09: Proceedings of the Twelfth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2009. IEEE Computer Society.

[23] C. Rother, V. Kolmogorov, and A. Blake. 'grabcut': interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[24] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010.

[25] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

[26] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis. Human detection using partial least squares analysis. In *International Conference on Computer Vision*, 2009.

[27] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27, 2010.

[28] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Techn.*, 18(11):1473–1488, 2008.

[29] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[30] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2010.

[31] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, pages 984–989, 2005.