Unsupervised Alignment of Natural Language Instructions with Video Segments

Iftekhar Naim, Young Chol Song, Qiguang Liu,
Henry Kautz, Jiebo Luo, Daniel Gildea
Department of Computer Science, University of Rochester
Rochester, NY 14627

Abstract

We propose an unsupervised learning algorithm for automatically inferring the mappings between English nouns and corresponding video objects. Given a sequence of natural language instructions and an unaligned video recording, we simultaneously align each instruction to its corresponding video segment, and also align nouns in each instruction to their corresponding objects in video. While existing grounded language acquisition algorithms rely on pre-aligned supervised data (each sentence paired with corresponding image frame or video segment), our algorithm aims to automatically infer the alignment from the temporal structure of the video and parallel text instructions. We propose two generative models that are closely related to the HMM and IBM 1 word alignment models used in statistical machine translation. We evaluate our algorithm on videos of biological experiments performed in wetlabs, and demonstrate its capability of aligning video segments to text instructions and matching video objects to nouns in the absence of any direct supervision.

Introduction

Learning to map natural language expressions to their corresponding referents in the physical environment is known as *grounded language acquisition*. Recently there has been growing interest in grounded language acquisition. The existing works typically assume the availability of aligned parallel data where each natural language sentence is paired with its corresponding image or video segment (Krishnamurthy and Kollar 2013; Tellex et al. 2013; Matuszek et al. 2012; Tellex et al. 2011). Manually pairing each video segment or image frame with the corresponding sentence can be tedious and may not be scalable to a large collection of videos and associated parallel text. In this paper, we aim to automatically align video frames with their corresponding natural language expressions without any direct supervision. We also jointly learn the correspondences between nouns in the sentences and their referents in the video.

We focus on the task of learning from the recorded videos of biological experiments performed in "wet laboratories". One of the key challenges in the biological sciences is to properly document experimental steps to ensure that results can be replicated following these documentations. A recent study reported that many major results in cancer biology are not reproducible (Begley and Ellis 2012). In this study, the researchers of Amgen Inc. attempted to replicate the results of 53 "landmark" cancer biology publications, and 43 of these 53 results could not be reproduced. Therefore, it is crucial to investigate and improve the standard of these documentations. Typically, each wetlab experiment has a protocol written in natural language, describing the sequence of steps necessary for that experiment. We take a set of such protocols, and collect videos of different people following these protocols and performing the experiments. Our initial goal is to infer the correct alignment between the steps mentioned in the protocol and corresponding video segments in which a person performs these steps (Figure 1). Eventually, we are interested in identifying experimental anomalies by using the aligned and segmented output of the system described in this paper to learn detailed visual models of correctly performed activities.

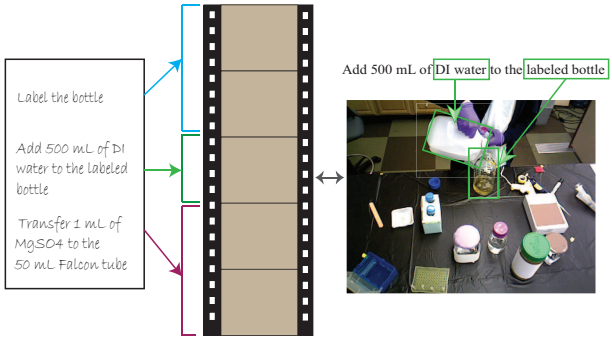


Figure 1: The proposed algorithm aligns protocol sentences to corresponding video frames, and simultaneously matches nouns in the sentences to corresponding blobs in video.

Each video frame is segmented into a set of objects (referred to as 'blobs'), and each blob is tracked over all the frames in the video. We perform a hierarchical alignment, where we align the protocol steps to their corresponding video segments where the person executes that step. We also

simultaneously align nouns to their corresponding blobs. The higher-level alignment is performed by a generative Hidden Markov Model (HMM) (Rabiner 1989; Vogel, Ney, and Tillmann 1996) that is similar to probabilistic dynamic time warping. For the lower level alignment/matching of blobs in the video to nouns, we use two different models: (1) IBM Model 1 (Brown et al. 1993) used in machine translation and (2) a novel extension of IBM Model 1 capable of handling missing nouns that are not observed in video. Our model is unsupervised and does not require any classifier to recognize the blobs in the video frames. Instead we treat the mapping between the nouns and corresponding video blobs as latent variables, and apply the Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977). The hierarchical alignment model used in this paper is similar to the models for aligning parallel corpora in machine translation (Brown, Lai, and Mercer 1991; Moore 2002), that perform alignment both at the word level and at the sentence level.

In this paper, we focus mainly on the alignment problem, and we do not perform semantic parsing of the sentences (Branavan et al. 2009; Vogel and Jurafsky 2010; Liang, Jordan, and Klein 2011). For alignment, we only perform syntactic parsing, and learn the mapping between nouns to blobs in the video frames. In the future, we plan to extend our model to learn the perceptual mappings for verbs, attributes, and other linguistic constituents.

Related Work

Grounded Language Acquisition

Learning the meaning representations of natural language constructs is an important problem in computational semantics. The meaning is typically represented either using symbolic logical forms (Zettlemoyer and Collins 2005; 2009) or by grounding them to entities, events, and relations in a physical environment specified by a segmented image or a video (Krishnamurthy and Kollar 2013; Tellex et al. 2013; Matuszek et al. 2012; Tellex et al. 2011). In this paper, we focus on the second type of meaning representation (known as grounded language acquisition). Tellex et al. (2011) proposed a probabilistic graphical model G^3 (Generalized Grounding Graphs) to infer the grounding of natural language constituents, given a set of natural language commands, each paired with the video example of the corresponding command being carried out. Matuszek et al. (2012) proposed a system that automatically learns the meaning of different attributes. The system is first trained using a fully supervised initial training stage, and learns a set of classifiers to perceive the initial set of attributes from the image features. Next, it incrementally learns the meaning of new attributes in a semi-supervised manner. For the initial supervised stage, each sentence needs to be paired with the corresponding image or video frame, and furthermore each individual object and attribute in the image needs to be labeled with its corresponding word or phrase in that sentence. Krishnamurthy et al. (2013) proposed the LSP (Logical Semantics with Perception) model, which jointly learns the semantic parsing of natural language sentences to logical forms and also the perceptual classifiers to recognize these concepts in the physical world. Both the LSP model (Krishnamurthy and Kollar 2013) and the G^3 model (Tellex et al. 2013) can treat the mapping between the language constituents and corresponding physical entities/relations as latent correspondence variables, and thus can avoid fully supervised training. However, these algorithms still need to know the exact pairing of the natural language sentences with their corresponding examples (e.g., exact pairing of natural language commands with corresponding video segments of a robot carrying out the command, the pairing of database queries with the matching entities, etc.).

We propose a hierarchical alignment model that jointly infers the pairing between natural language sentences and the corresponding video frames and also learns the mapping between the noun phrases to the physical entities present in those video frames. Our model is similar to the hierarchical HMM model by Liang et al. (2009), applied for aligning natural language utterances to the corresponding fields in database-like records. We model the correspondence between nouns and blobs by IBM Model 1 for word alignment. We embed IBM Model 1 inside a Hidden Markov Model (HMM) to exploit the temporal structures in the video and corresponding text sequences. Our work is different from the existing works on grounded language acquisition in the following ways: (1) Unlike the existing methods, our model does not require any alignment or pairing between natural language sentences/commands and their corresponding video segments, (2) we apply it to the complex biological wetlab experiments which is an interesting and novel application domain, and (3) we do not need to learn any perceptual classifiers for the objects in our model. The alignment inferred by our model can be useful to generate training data for perceptual classifiers.

Translating Image Blobs to Words

The IBM word alignment models for machine translation have previously been applied in the context of object recognition (Duygulu et al. 2002; Duygulu, Batan, and Forsyth 2006; Wachsmuth, Stevenson, and Dickinson 2003; Jamieson et al. 2006). Duygulu et al. (2002; 2006) applied IBM Model 1 and Model 2 to learn the correspondence between image regions and associated word tokens, given a large parallel corpus of images, each annotated with a set of key words. These images are first segmented into regions, and the features extracted from these regions are classified into a set of K visual words using K-means clustering. Finally, these visual words are aligned with their corresponding English keywords using IBM word alignment models. Wachsmuth et al. (2003) and later Jamieson et al. (2006) also applied Model 1 to align lower level image features (e.g. local SIFT features, shape features, etc.), instead of pre-segmented image regions. All these models rely on image-level annotated training data. Recently, several methods have been proposed for translating video content to natural language description (Krishnamoorthy et al. 2013; Yu and Siskind 2013; Rohrbach et al. 2013). These methods, however, were trained using pre-segmented short clips (less than 5 seconds long), each paired with a single sentence. We

Protocol	# Steps	# Sentences	Avg Video Length
CELL	13	34	7.78 minutes
LLGM	5	12	2.15 minutes
YPAD	8	25	4.14 minutes

Table 1: Statistics about the 3 protocols used in our experiments.

extend these models to longer videos and text sequences by embedding the IBM 1 model inside an HMM, relaxing the per-sentence annotation requirement. A similar problem has been addressed by Cour et al. (2008) to automatically align movie segments with the screen-play, but their solution relies on precisely time-aligned closed captions, which is not required for our method.

Video Segmentation and Object Tracking

For aligning text protocol with video, each blob needs to be detected and tracked by the vision system. Our work is similar to the work by Li et al. (2013) that detects hand-held objects in kitchen environment via CRF and tracks them via MeanShift tracker in RGB videos. We use RGB-D videos to achieve better performance (Song and Xiao 2013). Lei et al. (2012) proposed a system that tracks objects and hands in 2D space using RGB-D descriptors. Our approach is different in that it works in 3D space. In addition, we consider frequently present yet challenging transparent objects (e.g., glass bottles, jars, etc.), for which depth is usually zero and therefore their 3D positions are intractable.

Problem Overview

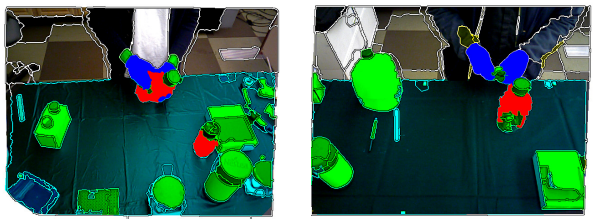
The input to our system is a video recording of a wetlab experiment accompanied with a protocol written in natural language describing the actions to be performed in that experiment. Our goal is to automatically align the video segments to the corresponding protocol sentences, and simultaneously learn the probabilities of matching each blob to the nouns in these sentences. We track hands in the input video, and consider only the blobs touched by hands.

Dataset Description

Our wetlab dataset has three different protocols: Cellobiose M9 Media (CELL), LB Liquid Growth Media (LLGM), and Yeast YPAD Media (YPAD). Each protocol consists of a sequence of instructions. Each sentence in the protocol either describes a high-level step or a sub-step that needs to be performed. Typically a step corresponds to one logical task unit in the experiment. The properties of each of the three protocols are presented in Table 1. For each protocol, we collect videos of several people executing the experiments. The videos are captured using HD video camera and an ASUS Xtion Pro RGB-Depth sensor.

Data Preprocessing

For detecting and tracking blobs in video, we first identify the workbench area in the 3D space by recovering the point clouds using the depth image of the RGB-D camera.



(a) CELL protocol (b) LLGM protocol

Figure 2: Two examples of object detection in wet lab, for two different experimental settings. Hands (blue) and objects (green) are above the table plane, and transparent objects are marked red.

Only areas above the workbench in the 3D space are considered as candidates for hands and objects. The scene is first segmented using an adjacency matrix representing the set of connected components that correspond to one or more continuous objects. Two points (p_1, p_2) in the point cloud are considered connected when their Euclidean distance is less than a designated threshold d_{thresh}. The cloud is further over-segmented by color using a modified version of the SLIC superpixel algorithm (Achanta et al. 2012). Then the superpixels are grouped using a greedy approach by their color and boundary map (Luo and Guo 2003). Features from color and 3D shape are used for segmentation of different objects, which are tracked using a 3D Kalman filter with pre-trained Gaussian Mixture Appearance Models. The interaction between hands and objects is inferred in 3D space. The transparent objects are detected using their zero depth property. We detect transparent objects in two stages. First, we filter out as many false positives as possible in individual frames using several rules: (1) transparent objects should either connect to the workbench or be connected by objects above the workbench, (2) small areas far away from the hands are filtered out, because small areas not occluded by hands are likely to be noise. In the second stage, we use a Kalman filter to track candidate areas based on their 2D position, size and speed. Noise areas with a short existence duration are filtered out.

Next we preprocess the sentences in the protocol text. We parse each sentence using the two-stage Charniak-Johnson syntactic parser (Charniak and Johnson 2005). For each noun phrases in the parse tree, we extract the head nouns and ignore other nouns. For example, for the noun phrase 'falcon tube', we ignore the word 'falcon' and only use the head noun 'tube'. We also apply simple heuristic rules that filter out spurious nouns that do not represent any object: ignore noun phrases that are either (1) object of the verb 'write' or (2) immediately preceding the word 'of'.

Challenges Faced

- *Unmentioned objects:* Some video segments have objects that are not mentioned in the corresponding text protocols. For example, protocol sentences like 'write X on the label" correspond to video segments where a person is touching a pen, but there is no noun that corresponds

to the pen in the protocol. Similarly, there are video segments where user is holding a pipette, but the corresponding sentence does not have the noun "pipette", and instead looks like "Aspirate X to Y".

- *Unobserved nouns:* The syntactic parsing system extracts some nouns that do not correspond to any objects/blobs in video (e.g., 'protocol', 'outside', 'anything', etc.). The alignment algorithm needs to be robust to these unobserved nouns.
- *Out of order execution:* Sometimes the experimenter touches objects out of order, mostly to set up the ingredients before executing a high-level experiment step or to clean up after a step.
- *Object Detection:* Thin objects like spatulas and plastic boats are difficult to detect using the state of the art computer vision algorithms. The tracking for several objects was noisy, especially due to illumination variation that confounded the appearance model.

Joint Alignment and Matching

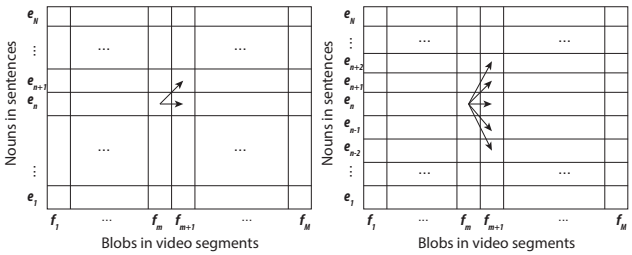
Input Representation

The object detection and tracking system from the previous section identifies the objects touched by hands in each video frame. The input video is split into small chunks, each one second long. For each video chunk, we identify the set of blobs touched by the hands during that time interval. We ignore the chunks over which no blob is touched by the hands. Finally, we get a sequence of video chunks $F = [\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(M)}]$, where each chunk $\mathbf{f}^{(m)} = \{f_1^{(m)}, \ldots, f_J^{(m)}\}$ is the set of blobs touched by the hands during that time interval. We extract head nouns from each of the protocol sentences, and represent the protocol text as a sequence of sets $E = [\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(N)}]$, where $\mathbf{e}^{(n)} = \{e_1^{(n)}, \ldots, e_I^{(n)}\}$ is the set of nouns in the n^{th} sentence.

Proposed Model: HMM + IBM 1

Given the blobs from M video chunks $F = [\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(M)}]$, and the nouns from N sentences $E = [\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(N)}]$, we want to infer the alignment between the video chunks and sentences. For computational tractability, we allow each video chunk to be aligned to only one of the sentences, but multiple chunks can be aligned to the same sentence. Let the alignment be $\mathbf{a}_1^M = a_1, a_2, \ldots, a_M$, where $a_m = n$ indicates that the m^{th} video segment is aligned to the n^{th} sentence. We also simultaneously learn the matching probability table ($T = p(f|e)$), that refers to the probability that the video object or blob f corresponds to the English noun e. We treat these correspondences as latent variables, and infer them using the EM framework.

We propose a hierarchical generative model to infer the alignment and matching. First, we generate the video chunks from the sentences using a Hidden Markov Model (HMM). Each video chunk $\mathbf{f}^{(m)}$ is generated from one of the sentences denoted by $a_m = n$, where $n \in \{1, \ldots, N\}$. Next, we generate the blobs in $\mathbf{f}^{(m)}$ from the nouns in $\mathbf{e}^{(n)}$ using IBM model 1. The alignment variable a_m is the hidden state



(a) HMM 1 (b) HMM 2

Figure 3: The two versions of HMM transitions used in our experiments.

for m^{th} time step in our HMM model. We use IBM Model 1 probabilities as emission probabilities. If the alignment state for the m^{th} video segment $a_m = n$, the emission probability is IBM Model 1 probability $\mathrm{P}(\mathbf{f}^{(m)}|\mathbf{e}^{(n)})$ of aligning $\mathbf{f}^{(m)}$ with $\mathbf{e}^{(n)}$.

Let V_E be the noun vocabulary, i.e., the set of all the unique nouns extracted from the entire text, and V_F be the set of all individual blobs in the entire video. The parameters in our model consist of a matching probability table $T = \{p(f|e)\}$, representing the probability of observing the blob f given the noun e. The probability table T provides soft matching probabilities between the blobs and nouns. The probability of generating a set of blobs $\mathbf{f}^{(m)} = \{f_1^{(m)}, \ldots, f_J^{(m)}\}$ from the set of nouns $\mathbf{e}^{(n)} = \{e_1^{(n)}, \ldots, e_I^{(n)}\}$ according to IBM Model 1 is:

$$\mathrm{P}\left(\mathbf{f}^{(m)}|\mathbf{e}^{(n)}\right) = \frac{\epsilon}{(I)^J} \prod_{j=1}^{J} \sum_{i=1}^{I} p(f_j^{(m)}|e_i^{(n)}) \qquad (1)$$

Following the Markov assumption, the hidden alignment state $a_m = n$ depends on the alignment state for the previous video segment $a_{m-1} = n'$. The state transition probability $\mathrm{P}(a_m = n|a_{m-1} = n')$ is parametrized by the jump size between adjacent alignment points:

$$\mathrm{P}(a_m = n|a_{m-1} = n') = c(n - n') \qquad (2)$$

where $c(l)$ represents the probability of jumps of distance l. To simplify the notation, we will refer to the transition probabilities $\mathrm{P}(a_m = n|a_{m-1} = n')$ as $\mathrm{P}(n|n')$. We experiment with two versions of HMMs, that vary in the transitions they are allowed to make (Figure 3). If $a_m = n$, the HMM 1 model allows $a_{m+1} \in \{n, n+1\}$. The HMM 2 model is more flexible, and allows five possible transitions (Figure 3(b)).

We aim to jointly learn the alignment $\mathbf{a}_1^M$ and the matching probabilities T given the input data. We apply the EM algorithm for the learning task. The parameters in our model are the matching probability matrix T and the jump probabilities c, and the latent variables are $\mathbf{a}_1^M$.

- **Initialize:** initialize the probability table $T = \{p(f|e)\}$ uniformly, i.e., we set $p(f|e) = 1/|V_F|$ for all $f \in V_F$ and $e \in V_E$. We also initialize the jump probabilities uniformly.

- **E-step:** apply the forward-backward recursions. The initial state is $a_0 = 0$. Let $\alpha_m(n)$ be the forward probability, $\beta_m(n)$ be the backward probability at m^{th} video chunk and state $a_m = n$.
  - The forward recursion:

$$\alpha_m(n) = \left[\sum_{n' \in \text{Pred}(n)} \alpha_{m-1}(n') \text{P}(n|n') \right] \text{P}\left(\mathbf{f}^{(m)}|\mathbf{e}^{(n)}\right) \quad (3)$$

    where $\text{Pred}(n)$ is the set of predecessors of the state n.
  - The backward recursion:

$$\beta_m(n) = \sum_{n' \in \text{Succ}(n)} \beta_{m+1}(n') \text{P}(n'|n)\ \text{P}\left(\mathbf{f}^{(m+1)}|\mathbf{e}^{(n')}\right) \quad (4)$$

    where $\text{Succ}(n)$ is the set of successors of the state n.
  - The posterior probability of being at state $a_m = n$ at time step m, which is denoted by $\gamma_m(n)$:

$$\gamma_m(n) = \frac{\alpha_m(n)\ \beta_m(n)}{\sum_{n'} \alpha_m(n')\ \beta_m(n')} \quad (5)$$

  - The posterior probability of the state pair $(a_m = n, a_{m+1} = n')$ denoted by $\xi_m(n, n')$:

$$\xi_m(n, n') = \frac{\alpha_m(n)\ \text{P}(n'|n)\text{P}\left(\mathbf{f}^{(m+1)}|\mathbf{e}^{(n')}\right)\ \beta_{m+1}(n')}{\sum_{n'} \sum_{n}\ \alpha_m(n)\ \text{P}(n'|n)\text{P}\left(\mathbf{f}^{(m+1)}|\mathbf{e}^{(n')}\right)\ \beta_{m+1}(n')} \quad (6)$$

  - Finally, for each possible alignment pair $(\mathbf{f}^{(m)}, \mathbf{e}^{(n)})$ in our HMM model, we estimate the expected counts of aligning a blob $f_j^{(m)} \in \mathbf{f}^{(m)}$ to a noun $e_i^{(n)} \in \mathbf{e}^{(n)}$.

$$EC_{m,n}(f_j^{(m)}, e_i^{(n)}) = \frac{p(f_j^{(m)}|e_i^{(n)})}{\sum_i p(f_j^{(m)}|e_i^{(n)})} \quad (7)$$

- **M-step:** we re-estimate the matching probabilities and jump probabilities using the posterior probabilities $\gamma_m(n)$ and $\xi_m(n, n')$ estimated in E-step:
  - Re-estimate matching probability table:

$$p(f|e) = \frac{\sum_{m=0}^{M} \sum_{n=0}^{N} \gamma_m(n) EC_{m,n}(f, e)}{\sum_f \sum_{m=0}^{M} \sum_{n=0}^{N} \gamma_m(n) EC_{m,n}(f, e)} \quad (8)$$

  - Re-estimate jump probabilities:

$$c(l) = \frac{\sum_{m=0}^{M} \sum_{n,n'} \xi_m(n, n')\ I[(n' - n) = l]}{\sum_{m=0}^{M} \sum_{n,n'} \xi_m(n, n')} \quad (9)$$

The computational complexity of each EM iteration is $O(MNIJ)$ for M video chunks, N sentences, at most I nouns per a sentence, and at most J blobs per video chunk.

	HMM 1 + IBM 1	HMM 2 + IBM 1	HMM 1 + Unobs	HMM 2 + Unobs
Anvil	41.09	39.73	**47.94**	**47.94**
Vision	28.76	28.76	26.02	**30.13**

Table 3: Average matching accuracy (% of objects correctly paired with corresponding nouns) for both Anvil annotations and automated computer vision tracking data.

HMM + IBM 1 + Unobserved Nouns

The text protocols contain nouns that are not observed in the video. Additionally, the computer vision algorithms for video segmentation and tracking often fail to identify small and transparent objects (e.g., pipette, spatula, etc.). We extend IBM 1 model to explicitly model these unobserved nouns. For each noun $e \in \mathbf{e}^{(n)}$, we introduce a boolean latent variable o_e:

$$o_e = \begin{cases} 1, & \text{if } e \text{ corresponds to a blob } f \in \mathbf{f}^{(m)} \\ 0, & \text{otherwise} \end{cases}$$

For each noun $e \in \mathbf{e}^{(n)}$, the generative process first samples the latent observation variables o_e, and then generates blobs in $\mathbf{f}^{(m)}$ only from the nouns for which $o_e = 1$. We assume that these o_e variables are conditionally independent and follow a Bernoulli distribution. The joint distribution is defined as:

$$\begin{aligned} \text{P}\left(\mathbf{f}^{(m)}, \mathbf{o}^{(n)}|\mathbf{e}^{(n)}\right) &= P(\mathbf{o}^{(n)}|\mathbf{e}^{(n)}) P\left(\mathbf{f}^{(m)}|\mathbf{o}^{(n)}, \mathbf{e}^{(n)}\right) \\ &= \left[\prod_{e \in \mathbf{e}^{(n)}} P(o_e) \right] \\ &\quad \left[\frac{\epsilon}{\left(\sum_{e \in \mathbf{e}^{(n)}} o_e\right)^J} \prod_{j=1}^{J} \sum_{e \in \mathbf{e}^{(n)}, o_e = 1} p(f_j^{(m)}|e) \right]. \end{aligned} \quad (10)$$

The emission probability is the marginal probability of generating the set of blobs $\mathbf{f}^{(m)} = \{f_1^{(m)}, \ldots, f_J^{(m)}\}$ from the set of nouns $\mathbf{e}^{(n)} = \{e_1^{(n)}, \ldots, e_I^{(n)}\}$:

$$\text{P}\left(\mathbf{f}^{(m)}|\mathbf{e}^{(n)}\right) = \sum_{\mathbf{o}^{(n)}} \text{P}\left(\mathbf{f}^{(m)}, \mathbf{o}^{(n)}|\mathbf{e}^{(n)}\right) \quad (11)$$

The complexity of marginalizing over all possible values of $\mathbf{o}^{(n)}$ grows exponentially with the number of nouns in a sentence. Since we typically have 4 or fewer nouns in each sentence, we can exactly compute these probabilities.

The EM procedure for the new model remains similar, with a few modifications. In the E-step, we estimate the forward-backward probabilities using the same recursions, but the emission probability $\text{P}\left(\mathbf{f}^{(m)}|\mathbf{e}^{(n)}\right)$ is estimated using equation 11. We also estimate the expected counts of observing each possible values $\mathbf{o}^{(n)}$ for each alignment position $a_m = n$:

$$EC_{m,n}(\mathbf{o}^{(n)}) = \frac{\text{P}\left(\mathbf{f}^{(m)}, \mathbf{o}^{(n)}|\mathbf{e}^{(n)}\right)}{\sum_{\mathbf{o}^{(n)}} \text{P}\left(\mathbf{f}^{(m)}, \mathbf{o}^{(n)}|\mathbf{e}^{(n)}\right)} \quad (12)$$

Protocol	Video ID	HMM 1 + IBM 1		HMM 2 + IBM 1		HMM 1 + Unobs		HMM 2 + Unobs		Baseline	
		Anvil	Vision	Anvil	Vision	Anvil	Vision	Anvil	Vision	Anvil	Vision
CELL	video-1	74.25	32.73	78.27	33.33	88.93	39.34	87.12	49.25	48.89	40.84
	video-2	76.31	52.70	85.54	52.70	81.30	52.70	88.53	52.70	49.38	24.32
LLGM	video-1	68.64	78.78	68.64	78.78	68.64	78.78	68.64	81.81	65.25	59.09
	video-2	70.25	61.33	70.25	30.67	67.77	30.67	68.59	30.67	86.78	58.67
YPAD	video-1	90.1	84.31	90.10	91.20	90.1	90.2	90.1	89.21	80.73	39.21
	video-2	94.53	70.0	93.80	70.0	94.53	70.0	92.70	70.7	72.27	61.42
Weighted Average		79.53	56.24	82.99	54.66	84.92	56.49	**86.02**	**59.63**	60.97	42.87

Table 2: Alignment accuracy (% of video chunks aligned to the correct protocol step) for both Anvil annotations and computer vision tracking data. For weighted averaging, the accuracy for each video is weighted by the length of that video.

We also estimate the expected counts of observing each pair (f, e) at each alignment position $a_m = n$ and each possible values of $\mathbf{o}^{(n)}$. In the M-step, we re-estimate T and c by normalizing the expected counts. Additionally, we re-estimate the observation probabilities $\mathrm{P}(o_e)$ for all $e \in V_E$:

$$\mathrm{P}(o_e = 1) = \frac{\sum_{j=0}^{M}\sum_{n=0}^{N}\sum_{\mathbf{o}^{(n)}:o_e=1} EC_{m,n}(\mathbf{o}^{(n)})\,\gamma_m(n)}{\sum_{j=0}^{M}\sum_{n=0}^{N}\sum_{\mathbf{o}^{(n)}} EC_{m,n}(\mathbf{o}^{(n)})\,\gamma_m(n)} \quad (13)$$

The computational complexity of each EM iteration is $O(MN2^I IJ)$.

Results and Discussions

We perform experiments on six wetlab videos (three protocols, two videos per protocol). To compare the errors introduced by our alignment algorithm and automated video segmentation and tracking systems, we evaluate alignment and matching accuracy both using automatically segmented videos and hand annotated videos. We manually annotate each of the videos to specify the objects touched by the hands using the video annotation tool *Anvil* (Kipp 2012). We experiment with two different types of HMM transition models (Figure 3) and two different emission models (IBM model 1 and its extension for unobserved nouns), and thus obtain 4 different versions of our alignment algorithm. The alignment accuracy is measured by the percentage of video chunks aligned to the correct protocol step. We compare our models with a simple baseline, where we uniformly distribute the video chunks among the sentences such that each sentence is aligned to an equal number of chunks. The alignment results (Table 2) show that our algorithm outperforms the uniform baseline, both on Anvil annotations and on the output of the computer vision system. Our best results are obtained by the "HMM 2 + Unobserved nouns" model, that explicitly models unobserved nouns. The HMM 2 model performed better than HMM 1 on average, as it allows touching objects out of protocol order. The accuracy is relatively lower on the automated computer vision data than that for Anvil data, particularly for small and thin objects (e.g. spatula, plastic boat. etc.), which are not tracked reliably by the vision system. On average, 60.9% of the blobs were detected and tracked reliably.

The proposed algorithm also works well in matching video blobs to nouns (Table 3). We examined the mistakes made by the matching algorithm, and many of the mistakes looked reasonable. For example, the video blob for the pen was mapped to the word 'label' in protocol, because the pen is used to write on the label and there is no word for pen in the protocol. Similarly the blob for syringe is often mapped to 'filter' of the syringe, and the blob for water jug got mapped to the word 'sink' from where water should be collected.

Conclusion and Future Work

In this paper, we propose a novel unsupervised learning algorithm for jointly aligning natural language instructions to video segments and for matching individual nouns in those instructions to corresponding video objects. We show that the proposed algorithm works well for complex videos of biological wetlab experiments, and can infer the alignment by exploiting the step-by-step structure of these videos. The proposed video alignment algorithm is a general framework, which can be applied to other types of video/text pairs that have a similar step-by-step structure (e.g., cooking videos paired with recipes, educational videos of scientific experiments paired with instructions, movies paired with screenplays, etc.).

There are several scopes of future improvements. Currently we use only nouns, and ignore verbs and relations in protocol text. Some of the verbs correspond to distinct hand movement patterns (e.g., mix, aspirate, pour, write, etc.), and often co-occur with particular objects in our videos (e.g., 'write' often co-occurs with a pen, 'aspirate' co-occurs with a pipette, etc.). We would like to learn perceptual features and hand movement patterns associated with different verbs and infer the relations between these verbs and different video objects. Sometimes the protocol does not explicitly mention some words, but they can be inferred from the context. For example, we have instruction sequences like: "Label the bottle. Add 40 mL DI water.". Although the second sentence does not explicitly mention the word 'bottle', it is apparent that water needs to be added to the bottle. Applying context-dependent semantic parsing (Zettlemoyer and Collins 2009) may allow us to infer such implicit words and improve the quality of alignment and matching.

Acknowledgments

This work was supported by DoD SBIR Award N00014-12-C-0263, the Google Faculty Research Award, NSF Award 1012017 and 1319378, ONR Award N00014-11-10417, ARO Award W911NF-08-1-0242, and the Intel Science & Technology Center for Pervasive Computing (ISTC-PC).

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Susstrunk, S. 2012. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(11):2274–2282.

Begley, C. G., and Ellis, L. M. 2012. Drug development: Raise standards for preclinical cancer research. *Nature* 483(7391):531–533.

Branavan, S. R. K.; Chen, H.; Zettlemoyer, L. S.; and Barzilay, R. 2009. Reinforcement learning for mapping instructions to actions. In *ACL/AFNLP*, 82–90.

Brown, P. F.; Della Pietra, S. A.; Della Pietra, V. J.; and Mercer, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2):263–311.

Brown, P. F.; Lai, J. C.; and Mercer, R. L. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th ACL*, 169–176. ACL.

Charniak, E., and Johnson, M. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *ACL*.

Cour, T.; Jordan, C.; Miltsakaki, E.; and Taskar, B. 2008. Movie/script: Alignment and parsing of video and text transcription. In *Proceedings of the 10th European Conference on Computer Vision: Part IV*, ECCV '08, 158–171. Berlin, Heidelberg: Springer-Verlag.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society* 39(1):1–21.

Duygulu, P.; Batan, M.; and Forsyth, D. 2006. Translating images to words for recognizing objects in large image and video collections. In Ponce, J.; Hebert, M.; Schmid, C.; and Zisserman, A., eds., *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 258–276.

Duygulu, P.; Barnard, K.; Freitas, J. d.; and Forsyth, D. A. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, 97–112. Springer-Verlag.

Jamieson, M.; Dickinson, S.; Stevenson, S.; and Wachsmuth, S. 2006. Using language to drive the perceptual grouping of local image features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 2, 2102–2109.

Kipp, M. 2012. Anvil: A universal video research tool. *Handbook of Corpus Phonology. Oxford University Press*.

Krishnamoorthy, N.; Malkarnenkar, G.; Mooney, R.; Saenko, K.; and Guadarrama, S. 2013. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-13)*, volume 2013, 3.

Krishnamurthy, J., and Kollar, T. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Assoc. for Comp. Ling.* 10:193–206.

Lei, J.; Ren, X.; and Fox, D. 2012. Fine-grained kitchen activity recognition using rgb-d. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (Ubicomp)*, 208–211. ACM.

Li, Y., and Luo, J. 2013. Task-relevant object detection and tracking. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*.

Liang, P.; Jordan, M. I.; and Klein, D. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, 91–99. Association for Computational Linguistics.

Liang, P.; Jordan, M. I.; and Klein, D. 2011. Learning dependency-based compositional semantics. In *ACL*, 590–599.

Luo, J., and Guo, C. 2003. Perceptual grouping of segmented regions in color images. *Pattern Recognition* 36(12):2781 – 2792.

Matuszek, C.; Fitzgerald, N.; Zettlemoyer, L.; Bo, L.; and Fox, D. 2012. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML-2012)*, 1671–1678.

Moore, R. C. 2002. Fast and accurate sentence alignment of bilingual corpora. In *AMTA '02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, 135–144. London, UK: Springer-Verlag.

Rabiner, L. R. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–286.

Rohrbach, M.; Qiu, W.; Titov, I.; Thater, S.; Pinkal, M.; and Schiele, B. 2013. Translating video content to natural language descriptions. In *14th IEEE International Conference on Computer Vision (ICCV)*, 433–440.

Song, S., and Xiao, J. 2013. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *14th IEEE International Conference on Computer Vision (ICCV 2013)*. IEEE.

Tellex, S. A.; Kollar, T. F.; Dickerson, S. R.; Walter, M. R.; Banerjee, A.; Teller, S.; and Roy, N. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-11)*. AAAI Publications.

Tellex, S.; Thaker, P.; Joseph, J.; and Roy, N. 2013. Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning* 1–17.

Vogel, A., and Jurafsky, D. 2010. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 806–814.

Vogel, S.; Ney, H.; and Tillmann, C. 1996. HMM-based word alignment in statistical translation. In *COLING-96*, 836–841.

Wachsmuth, S.; Stevenson, S.; and Dickinson, S. 2003. Towards a framework for learning structured shape models from text-annotated images. In *Proceedings of the HLT-NAACL 2003 Workshop on Learning Word Meaning from Non-linguistic Data - Volume 6*, HLT-NAACL-LWM '04, 22–29. Stroudsburg, PA, USA: Association for Computational Linguistics.

Yu, H., and Siskind, J. M. 2013. Grounded language learning from video described with sentences. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-13)*, volume 1, 53–63.

Zettlemoyer, L. S., and Collins, M. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*, 658–666.

Zettlemoyer, L. S., and Collins, M. 2009. Learning context-dependent mappings from sentences to logical form. In *ACL/AFNLP*, 976–984.