Unsupervised Semantic Parsing of Video Collections

Ozan Sener[1,2]   Amir R. Zamir[1]   Silvio Savarese[1]   Ashutosh Saxena[2,3]

[1]Stanford University, [2]Cornell University, [3]Brain of Things Inc.

{ozansener,zamir,ssilvio,asaxena}@cs.stanford.edu

Abstract

Human communication typically has an underlying structure. This is reflected in the fact that in many user generated videos, a starting point, ending, and certain objective steps between these two can be identified. In this paper, we propose a method for parsing a video into such semantic steps in an unsupervised way. The proposed method is capable of providing a "semantic storyline" of the video composed of its objective steps. We accomplish this using both visual and language cues in a joint generative model. The proposed method can also provide a textual description for each of the identified semantic steps. We evaluate this method on a large number of complex YouTube videos and show results of unprecedented quality for this intricate and impactful problem.

1. Introduction

Human communication takes many forms, including language and vision. For instance, explaining "how-to" perform a certain task can be communicated via language (*e.g.*, Do-It-Yourself books) as well as visual (*e.g.*, instructional YouTube videos) information. Regardless of the form, such human-generated communication is generally structured and has a clear beginning, end, and a set of steps in between. Parsing such communication into its semantic steps is the key to understanding structured human activities.

Language and vision provide different, but correlating and complementary information. Challenge lies in that both video frames and language (from subtitles generated via ASR) are only a noisy, partial observation of the actions being performed. However, the complementary nature of language and vision gives the opportunity to understand the activities from these partial observations. In this paper, we present a unified model, incorporating both of the modalities, in order to parse human activities into activity steps with no form of supervision other than requiring videos to be of the same category (*e.g.*, videos retrieved by query cooking eggs, changing tires, etc.).

The key idea in our approach is the observation that the large collection of videos, pertaining to the same activity
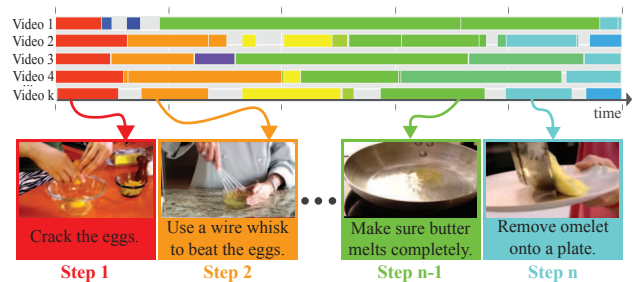

Figure 1: Given a large video collection (frames and subtitles) of an structured category (*e.g.*, How to cook an omelette?), we discover activity steps (*e.g.*, crack the eggs). We also parse the videos based on the discovered steps.

class, typically include only a few objective activity steps, and the variability is the result of exponentially many ways of generating videos from activity steps through subset selection and time ordering. We study this construction based on the large-scale information available in YouTube in the form of instructional videos (*e.g.*, "Making pancake", "How to tie a bow tie"). We adopt Instructional videos since they have many desirable properties like the volume of the information (*e.g.*, YouTube has 281.000 videos for *"How to tie a bow tie"*) and a well defined notion of activity step. However, the proposed parsing method is applicable to any type of structured videos as long as they are composed of a set of objective steps.

The output of our method can be seen as the "semantic storyline" of a rather long and complex video collection (see Fig. 1). This storyline provides what particular steps are taking place in the video collection, when they are occurring, and what their meaning is (*what-when-how*). This method also puts videos performing the same overall task in common ground and capture their high-level relations.

In the proposed approach, given a collection of videos, we first generate a set of language and visual atoms. These atoms are the result of relating object proposals from each frame as well as detecting the frequent words from subtitles. We then employ a generative *beta process mixture model*, which identifies the activity steps shared among the videos of the same category based on a representation us-

ing learned atoms. The discovered steps are found to be highly correlating with semantic steps since the semantics are the strongest common structure among all of the videos of one category. In our method, we use neither any spatial or temporal label on actions/steps nor any labels on object categories. We later learn a Markov language model to provide a textual description for each of the activity steps based on the language atoms it frequently uses.

2. Related Work

Three aspects differentiate this work from the majority of existing techniques: 1) discovering semantic steps from a video category, 2) being unsupervised, 3) adopting a multi-modal joint vision-language model for video parsing. A thorough review of the related literature is provided below.

Video Summarization: Summarizing an input video as a sequence of key frames (static) or video clips (dynamic) is useful for both multimedia search interfaces and retrieval purposes. Early works in the area are summarized in [59] and mostly focus on *choosing keyframes* for visualization.

Summarizing videos is particularly important for long sequences like ego-centric videos and news reports [35, 38, 50]; however, these methods mostly rely on characteristics of the application and do not generalize.

Summarization is also applied to the large image collections by recovering the temporal ordering and visual similarity of images [26], and by Gupta et al. [17] to videos in a supervised framework using annotations of actions. These collections are also used to choose important scenes for key-frame selection [24] and further extended to video clip selection [25, 48]. Unlike all of these methods which focus on forming a set of key frames/clips for a compact summary (which is not necessarily semantically meaningful), we provide a fresh approach to video summarization by performing it through semantic parsing on vision and language. However, regardless of this dissimilarity, we experimentally compare our method against them.

Modeling Visual and Language Information: Learning the relationship between the visual and language data is a crucial problem due to its immense applications. Early methods [4] in this area focus on learning a common multi-modal space in order to jointly represent language and vision. They are further extended to learning higher level relations between object segments and words [54]. Similarly, Zitnick et al. [63, 62] used abstracted clip-arts to understand spatial relations of objects and their language correspondences. Kong et al. [28] and Fidler et al. [13] both accomplished the task of learning spatial reasoning using the image captions. Relations extracted from image-caption pairs, are further used to help semantic parsing [61] and activity recognition [41]. Recent works also focus on automatic generation of image captions with underlying ideas ranging from finding similar images and transferring their captions [45] to learning language models conditioned on the image

features [27, 55, 12]; their employed approach to learning language models is typically either based on graphical models [12] or neural networks [55, 27, 23].

All aforementioned methods use supervised labels either as strong image-word pairs or weak image-caption pairs, while our method is fully unsupervised.

Activity/Event Recognition: The literature of activity recognition is broad. The closest techniques to ours are either supervised or focus on detecting a particular (and often short) action in a weakly/unsupervised manner. Also, a large body of action recognition methods are intended for trimmed videos clips or remain limited to detecting very short actions [30, 56, 42, 33, 11, 51]. Even though some recent works attempted action recognition in untrimmed videos [21, 44, 20], they are mostly fully supervised.

Additionally, several method for localizing instances of actions in rather longer video sequences have been developed [10, 18, 34, 6, 47]. Our work is different from those in terms of being multimodal, unsupervised, applicable to a video collection, and not limited to identifying predefined actions or the ones with short temporal spans. Also, the previous works on finding action primitives such as [42, 60, 19, 32, 31] are primarily limited to discovering atomic sub-actions, and therefore, fail to identify complex and high-level parts of a long video.

Recently, event recounting has attracted much interest and intends to identify the evidential segments for which a video belongs to a certain class [57, 9, 3]. Event recounting is a relatively new topic and the existing methods mostly employ a supervised approach. Also, their end goal is to identify what parts of a video are highly related to an event, and not parsing the video into semantic steps.

Recipe Understanding: Following the interest in community generated recipes in the web, there have been many attempts to automatically process recipes. Recent methods on natural language processing [40, 58] focus on semantic parsing of language recipes in order to extract actions and the objects in the form of predicates. Tenorth et al. [58] further process the predicates in order to form a complete logic plan. The aforementioned approaches focus only on the language modality and they are not applicable to the videos. The recent advances [5, 7] in robotics use the parsed recipe in order to perform cooking tasks. They use supervised object detectors and report a successful autonomous experiment. In addition to the language based approaches, Malmaud et al. [39] consider both language and vision modalities and propose a method to align an input video to a recipe. However, this method can not extract the steps automatically and requires a ground truth recipe to align. On the contrary, our method uses both visual and language modalities and extracts the actions while autonomously discovering the steps. Also, [15] generates multi-modal recipes from expert demonstrations . However, it is developed only for the domain of "teaching user interfaces" and are not ap-

plicable to videos.

3. Overview

Given a large video-collection, our algorithm starts with learning a set of visual and language atoms which are further used for representing multimodal information (Section 4). These atoms are designed to be more likely to correspond to the mid-level semantic concepts like actions and objects. In order to learn visual atoms, we generate object proposals and cluster them into mid-level atoms. Whereas, for the language atoms we simply use the salient and frequent words in the subtitles. After learning the atoms, we represent the multi-modal information in each frame based on the occurrence statistics of the atoms (Section 4); Given the sequence of multi-modal frame representations, we discover a set of clusters occurring over multiple videos using a non-parametric Bayesian method (Section 5.1). We expect these clusters to correspond to the activity steps which construct the high level activities. Our empirical results confirms this as the resulting clusters significantly correlates with the activity steps.

4. Forming the Multi-Modal Representation

Finding the set of activity steps over large collection of videos having large visual varieties requires us to represent the semantic information in addition to the low-level visual cues. Hence, we find our language and visual atoms by using mid-level cues like object proposals and frequent words.
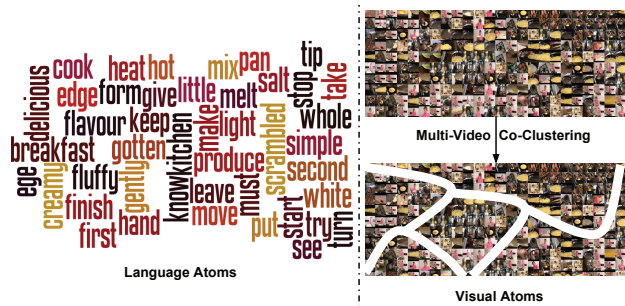

Figure 2: We learn language and visual atoms to represent multi-modal information. Language atoms are frequent words and visual atoms are the clusters of object proposals.

Learning Visual Atoms: In order to learn visual atoms, we create a large collection of proposals by independently generating object proposals from each frame of each video. These proposals are generated using the Constrained Parametric Min-Cut (CPMC) [8] algorithm based on both appearance and motion cues. We note the kth proposal of tth frame of ith video as $r_t^{(i),k}$. Moreover, we drop the video index (i) if it is clearly implied in the context.

In order to group this object proposals into mid-level visual atoms, we follow a clustering approach. Although any graph clustering approach (*e.g.*, Keysegments [36]) can be applied for this, the joint processing of a large video collec-

tion requires handling *large visual variability* among multiple videos. We propose a new method to jointly cluster object proposals over multiple videos in Section 5. Each cluster of object proposals correspond to a visual atom.

Learning Language Atoms: We define the language atoms as the salient words which occur more often than their ordinary rates based on the *tf-idf* measure. The *document* is defined as the concatenation of all subtitles of all frames of all videos in the collection. Then, we follow the classical tf-idf measure and use it as $tfidf(w, D) = f_{w,D} \times \log\left(1 + \frac{N}{n_w}\right)$ where w is the word we are computing the tf-idf score for, $f_{w,D}$ is the frequency of the word in the *document* D, N is the total number of video collections we are processing, and n_w is the number of video collections whose subtitle include the word w.

We sort words with their "tf-idf" values and choose the top K words as language atoms ($K = 100$ *in our experiments*). As an example, we show the language atoms learned for the category *making scrambled egg* in Figure 2.

Representing Frames with Atoms: After learning the visual and language atoms, we represent each frame via the occurrence of atoms (binary histogram). Formally, the representation of the tth frame of the ith video is denoted as $\mathbf{y_t^{(i)}}$ and computed as $\mathbf{y_t^{(i)}} = [\mathbf{y_t^{(i),l}}, \mathbf{y_t^{(i),v}}]$ such that k^{th} entry of the $\mathbf{y_t^{(i),l}}$ is 1 if the subtitle of the frame has the kth language atom and 0 otherwise. $\mathbf{y_t^{(i),v}}$ is also a binary vector similarly defined over visual atoms. We visualize the representation of a sample frame in the Figure 3.
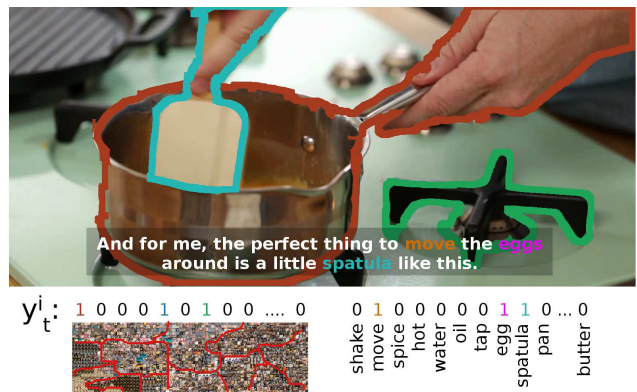

Figure 3: **Representation for a sample frame.** Three of the object proposals of sample frame are in the visual atoms and three of the words are in the language atoms.

5. Joint Proposal Clustering over Videos

Given a set of object proposals generated from *multiple videos*, simply combining them into a single collection and clustering them into atoms is not desirable for two reasons: (1) semantic concepts have large visual differences among different videos and accurately clustering them into a single atom is hard, (2) atoms should contain object proposals from multiple videos in order to semantically relate the

videos. In order to satisfy these requirements, we propose a joint extension to spectral clustering. Note that the purpose of this clustering is generating atoms where each clusters represents an atom.
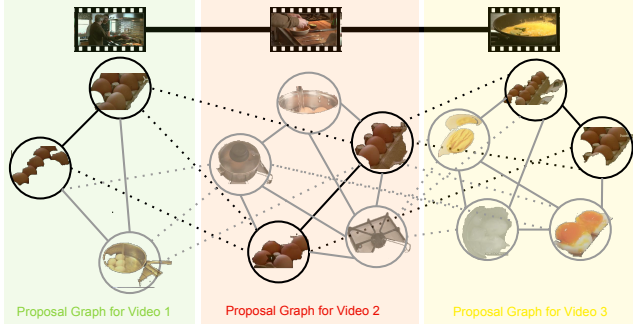

Figure 4: **Joint proposal clustering.** Each object proposal is linked to its two NNs from the video it belongs and two NNs from the videos it is neighbour of. Dashed and solid lines denote the intra-video and inter-video edges, respectively. Black nodes are the proposals selected as part of the cluster and the gray ones are not selected. Similarly, the black and gray edges denote selected and not-selected, respectively.

Basic Graph Clustering: Consider the set of object proposals extracted from a single video $\{r_t^k\}$, and a pairwise similarity metric $d(\cdot, \cdot)$ for them. Single cluster graph partitioning (SCGP)[43] approach finds the dominant cluster which maximizes the intra-cluster similarity:

$$\underset{x_t^k}{\arg\max} \frac{\sum_{(k_1,t_1),(k_2,t_2)\in K\times T} x_{t_1}^{k_1} x_{t_2}^{k_2} d(r_{t_1}^{k_1}, r_{t_2}^{k_2})}{\sum_{(k,t)\in K\times T} x_t^k}, \quad (1)$$

where x_t^k is a binary variable which is 1 if r_t^k is included in the cluster, T is the number of frames and K is the number of clusters per frame. Adopting the vector form of the indicator variables as $\mathbf{x_{tK+k}} = x_t^k$ and the pairwise distance matrix as $\mathbf{A}_{t_1K+k_1,t_2K+k_2} = d(r_{t_1}^{k_1}, r_{t_2}^{k_2})$, equation (1) can be compactly written as $\arg\max_{\mathbf{x}} \frac{\mathbf{x^T}A\mathbf{x}}{\mathbf{x^T}\mathbf{x}}$. This can be solved by finding the dominant eigenvector of $\mathbf{x}$ after relaxing x_t^k to $[0, 1]$ [43, 46]. Upon finding the cluster, the members of the selected cluster are removed from the collection and the same algorithm is applied to find remaining clusters.

Joint Clustering: Our extension of the SCGP into multiple videos is based on the assumption that the key objects occur in most of the videos. Hence, we re-formulate the problem by enforcing the homogeneity of the cluster over all videos.

We first create a kNN graph of the videos based on the distance between their textual descriptions. We use the χ^2 distance of the bag-of-words computed from the video description. We also create the kNN graph of object proposals in each video based on the pretrained "fc7" features of AlexNet [29]. This hierarchical graph structure is visualized in Figure 4 for three videos samples. After creating this

graph, we impose both "inter-video" and "intra-video" similarity among the object proposals of each cluster. Main rationale behind this construction is having a separate notion of distance for inter-video and intra-video relations since the visual similarity decreases drastically for inter-video ones.

Given the intra-video distance matrices $\mathbf{A}^{(i)}$, the binary indicator vectors $\mathbf{x}^{(i)}$, and the inter-video distance matrices as $\mathbf{A}^{(i,j)}$, we define our optimization problem as:

$$\arg\max \sum_{i\in N} \frac{\mathbf{x}^{(i)T}\mathbf{A}^{(i)}\mathbf{x}^{(i)}}{\mathbf{x}^{(i)T}\mathbf{x}^{(i)}} + \sum_{i\in N}\sum_{j\in\mathcal{N}(i)} \frac{\mathbf{x}^{(i)T}\mathbf{A}^{(i,j)}\mathbf{x}^{(j)}}{\mathbf{x}^{(i)T}\mathbb{1}\mathbb{1}^T\mathbf{x}^{(j)}}, \tag{2}$$

where $\mathcal{N}(i)$ is the neighbours of the video i in the kNN graph, $\mathbb{1}$ is vector of ones and N is the number of videos.

Although we can not use the efficient eigendecomposition approach from [43, 46] as a result of the modification, we can use Stochastic Gradient Descent as the cost function is quasi-convex when relaxed. We use the SGD with the following analytic gradient function:

$$\nabla_{\mathbf{x}^{(i)}} = \frac{2\mathbf{A}^{(i)}\mathbf{x}^{(i)} - 2\mathbf{x}^{(i)}r^{(i)}}{\mathbf{x}^{(i)T}\mathbf{x}^{(i)}} + \sum_{i\in N} \frac{\mathbf{A}^{i,j}\mathbf{x^j} - \mathbf{x}^{(j)T}\mathbb{1}r^{(i,j)}}{\mathbf{x}^{(i)T}\mathbb{1}\mathbb{1}^T\mathbf{x}^{(j)}}, \tag{3}$$

where $r^{(i)} = \frac{\mathbf{x}^{(i)T}\mathbf{A}^{(i)}\mathbf{x}^{(i)}}{\mathbf{x}^{(i)T}\mathbf{x}^{(i)}}$ and $r^{(i,j)} = \frac{\mathbf{x}^{(i)T}\mathbf{A}^{(i,j)}\mathbf{x}^{(j)}}{\mathbf{x}^{(i)T}\mathbb{1}\mathbb{1}^T\mathbf{x}^{(j)}}$

We iteratively use the method to find clusters, and stop after the $K = 20$ clusters are found as the remaining object proposals were deemed not relevant to the activity. Each cluster corresponds to a visual atom for our application.

In Figure 5, we visualize some of the atoms (*i.e.*, clusters) we learned for the query *How to Hard Boil an Egg?*. As apparent in the figure, the resulting atoms are highly correlated and correspond to semantic objects&concepts regardless of their significant intra-class variability.
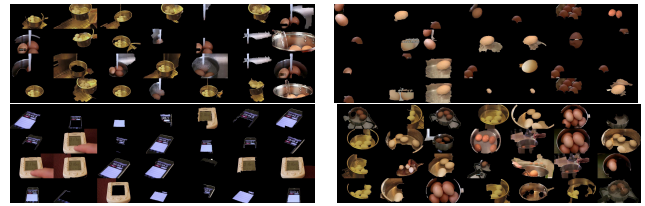

Figure 5: Randomly selected images of four randomly selected clusters learned for *How to hard boil an egg?*

5.1. Unsupervised Parsing

In this section, we explain the model which we use to discover the activity steps from a video collection given the language and visual atoms. We note the extracted representation of the frame t of video i as $\mathbf{y_t^{(i)}}$. We model our algorithm based on activity steps and note the activity label of the tth frame of the ith video as $z_t^{(i)}$. We do not fix the the number of activities and use a non-parametric approach.

In our model, each activity step is represented over the atoms as the likelihood of including them. In other words, each activity step is a Bernoulli distribution over the visual and language atoms as $\theta_k = [\theta_k^l, \theta_k^v]$ such that m^{th} entry of the θ_k^l is the likelihood of observing m^{th} language atom in the frame of an activity k. Similarly, m^{th} entry of the θ_k^v represents the likelihood of seeing m^{th} visual atom. In other words, each frame's representation $\mathbf{y_t^{(i)}}$ is sampled from the distribution corresponding to its activity as $\mathbf{y_t^{(i)}}|z_t^{(i)} = k \sim Ber(\theta_k)$. As a prior over θ, we use its conjugate distribution – *Beta distribution*.

Given the model above, we explain the generative model which links activity steps and frames in Section 5.1.1.

5.1.1 Beta Process Hidden Markov Model

For the understanding of the time-series information, Fox et al. [14] proposed the Beta Process Hidden Markov Models (BP-HMM). In BP-HMM setting, each time-series exhibits a subset of available features. Similarly, in our setup each video exhibits a subset of activity steps.

Our model follows the construction of Fox et al. [14] and differs in the choice of probability distributions since [14] considers Gaussian observations while we adopt binary observations of atoms. In our model, each video i chooses a set of activity steps through an activity step vector $\mathbf{f^{(i)}}$ such that $f_k^{(i)}$ is 1 if i^{th} video has the activity step k, and 0 otherwise. When the activity step vectors of all videos are concatenated, it becomes an activity step matrix $\mathbf{F}$ such that i^{th} row of the $\mathbf{F}$ is the activity step vector $\mathbf{f^{(i)}}$. Moreover, each activity step k also has a prior probability b_k and a distribution parameter θ_k which is the Bernoulli distribution as we explained in the Section 5.1.

In this setting, the activity step parameters θ_k and b_k follow the *beta process* as;

$$B|B_0, \gamma, \beta \sim \text{BP}(\beta, \gamma B_o), B = \sum_{k=1}^{\infty} b_k \delta_{\theta_k} \quad (4)$$

where B_0 and the b_k are determined by the underlying Poisson process [16] and the feature vector is determined as independent Bernoulli draws as $f_k^{(i)} \sim Ber(b_k)$. After marginalizing over the b_k and θ_k, this distribution is shown to be equivalent to Indian Buffet Process (IBP) [16]. In the IBP analogy, each video is a customer and each activity step is a dish in the buffet. The first customer (video) chooses a Poisson(γ) unique dishes (activity steps). The following customer (video) i chooses previously sampled dish (activity step) k with probability $\frac{m_k}{i}$, proportional to the number of customers (m_k) chosen the dish k, and it also chooses Poisson($\frac{\gamma}{i}$) new dishes (activity steps). Here, γ controls the number of selected activities in each video and β promotes the activities getting shared by videos.

The above IBP construction represents the activity step discovery part of our method. In addition, we need to model

the video parsing over discovered steps; these two need to be modeled jointly. We model the each video as an Hidden Markov Model (HMM) over the selected activity steps. Each frame has the hidden state –activity step– ($z_t^{(i)}$) and we observe the multi-modal frame representation $\mathbf{y_t^{(i)}}$. Since we model each activity step as a Bernoulli distribution, the emission probabilities follow the Bernoulli distribution as $p(\mathbf{y_t^{(i)}}|z_t^{(i)}) = Ber(\theta_{z_t^{(i)}})$.

For the transition probabilities of the HMM, we do not put any constraint and simply model it as any point from a probability simplex which can be sampled by drawing a set of Gamma random variables and normalizing them [14]. For each video i, a Gamma random variable is sampled for the transition between activity step j and activity step k if both of the activity steps are included in the video (*i.e.* if f_k^i and f_j^i are both 1). After sampling these random variables, we normalize them to make transition probabilities to sum to 1. This procedure can be represented formally as

$$\eta_{j,k}^{(i)} \sim Gam(\alpha + \kappa\delta_{j,k}, 1), \quad \pi_{\mathbf{j}}^{(\mathbf{i})} = \frac{\eta_{\mathbf{j}}^{(\mathbf{i})} \circ \mathbf{f^{(i)}}}{\sum_k \eta_{j,k}^{(i)} f_k^{(i)}}, \quad (5)$$

where κ is the persistence parameter promoting the self state transitions a.k.a. more coherent temporal boundaries, $\circ$ is the element-wise product, and π_j^i is the transition probabilities in video i from activity step j to other steps. This model is also presented as a graphical model in Figure 6.
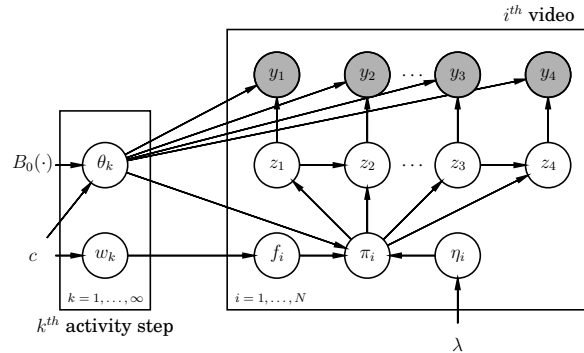

Figure 6: **Graphical model for BP-HMM:** The left plate represent the activity steps and the right plate represent the videos. (*i.e.* the left plate is for the activity step discovery and right plate is for parsing.) *See Section 5.1.1 for details.*

5.1.2 Gibbs sampling for BP-HMM

We employ Markov Chain Monte Carlo (MCMC) method for learning and inference of the BP-HMM. We base our algorithms on the MCMC procedure proposed by Fox et al. [14]. Our sampling procedure composed of two samplers: (1) activity step ($\mathbf{f^{(i)}}$) sampler from the current activity step distributions $\theta_\mathbf{k}$ and multi-modal frame representations $y_k^{(i)}$, (2) and HMM parameter $\eta, \pi, \theta_\mathbf{k}$ sampler from the

selected activities $\mathbf{f}^{(\mathbf{i})}$. Intuitively, we iterate over discovering activity steps given the temporal activity labels and estimating activity labels given the discovered activities. We give the details of this sampler in [1].

6. Experiments

In order to experiment the proposed method, we first collected a dataset (details in Section 6.1). We labelled a small part of the dataset with frame-wise activity step labels and used the resulting set as a test corpus. Neither the set of labels, nor the temporal boundaries are exposed to our algorithm since the setup is completely unsupervised. We evaluate our algorithm against the several unsupervised clustering baselines and state-of-the-art algorithms from video summarization literature which are applicable.

6.1. Dataset

We use WikiHow [2] to obtain the top100 queries the internet users are interested in and choose the ones which are related to the physical world. Resulting queries are;

How to Bake Boneless Skinless Chicken, Tie a Tie, Clean a Coffee Maker, Make Jello Shots, Cook Steak, Bake Chicken Breast, Hard Boil an Egg, Make Yogurt, Make a Milkshake, Make Beef Jerky, Make Scrambled Eggs, Broil Steak, Cook an Omelet, Make Ice Cream, Make Pancakes, Remove Gum from Clothes, Unclog a Bathtub Drain

For each of the queries, we crawled YouTube and got the top 100 videos. We also downloaded the English subtitles if they exist. For the test set, we randomly choose 5 videos out of 100 per query.

6.1.1 Outlier Detection

Since we do not have any expert intervention in our data collection, the resulting collection might have outliers, mainly due to fact that our queries are typical daily activities and there are many cartoons, funny videos, and music videos about them. Hence, we have an automatic coarse filtering stage. The key-idea behind the filtering algorithm is the fact that instructional videos have a distinguishable text descriptions when compared with outliers. Hence, we use a clustering algorithm to find the dominating cluster of instructional videos free of outliers. Given a large video collection, we use the graph, explained in Section 5, and compute the dominant video cluster by using the Single Cluster Graph Partitioning [43] and discards the remaining videos as outlier. In Figure 7, we visualize some of the discarded videos. Although our algorithm have a few percentage of false positives while detecting outliers, we always have enough number of videos (minimum 50) after the outlier detection, thanks to the large-scale dataset.

6.2. Qualitative Results

After independently running our algorithm on all categories, we discover activity steps and parse the videos according to discovered steps. We visualize some of these cat-

Figure 7: **Sample videos which our algorithm discards as an outlier for various queries.** A toy milkshake, a milkshake charm, a funny video about How to NOT make smoothie, a video about the danger of a fire, a cartoon video, a neck-tie video erroneously labeled as bow-tie, a song, and a lamb mislabeled as chicken.

egories qualitatively in Figure 8 with the temporal parsing of evaluation videos as well the ground truth parsing.

To visualize the content of each activity step, we display key-frames from different videos. We also train a 3rd order Markov language model [53] using the subtitles and employ it to generate a caption for each step by sampling this model conditioned on the θ_k^l. We explain the details in [1].

As shown in the Figures 8a and 8b, resulting steps are semantically meaningful; hence, we conclude that there is enough language context within the subtitles in order to detect activities. However, some of the activity steps occur together and our algorithm merges them into a single step as a result of promoting sparsity.

6.3. Quantitative Results

We compare our algorithm with the following baselines.
Low-level features (LLF): In order to experiment the effect of learned atoms, we compare them low-level features. As features, we use the Fisher vector representation of Dense Trajectory like features (HOG, HOF, and MBH) [22].
Single modality: To experiment the effect of multi-modal approach, we compare with single modalities by only using the atoms of one modality.
**Hidden Markov Model (HMM):** To experiment the effect of joint generative model, we compare our algorithm with an HMM (using the Baum-Welch [49] via cross-validation).
Kernel Temporal Segmentation [48]: Kernel Temporal Segmentation (KTS) proposed by Potapov et al. [48] can detect the temporal boundaries of the events/activities in the video from a time series data without any supervision. It enforces a local similarity of each resultant segment.

Given parsing results and the ground truth, we evaluate both the quality of temporal segmentation and the activity step discovery. We base our evaluation on two widely used metrics; intersection over union (**IOU**) and mean average precision(mAP). **IOU** measures the quality of temporal segmentation and it is defined as; $\frac{1}{N}\sum_{i=1}^{N}\frac{\tau_i^\star \cap \tau_i'}{\tau_i^\star \cup \tau_i'}$ where N is the number of segments, $\tau_i^\star$ is ground truth segment and τ_i' is the detected segment. **mAP** is defined per activity step and can be computed based on a precision-recall curve [21]. In order to adopt these metrics into unsupervised setting, we use cluster similarity measure(csm)[37] which enables us to
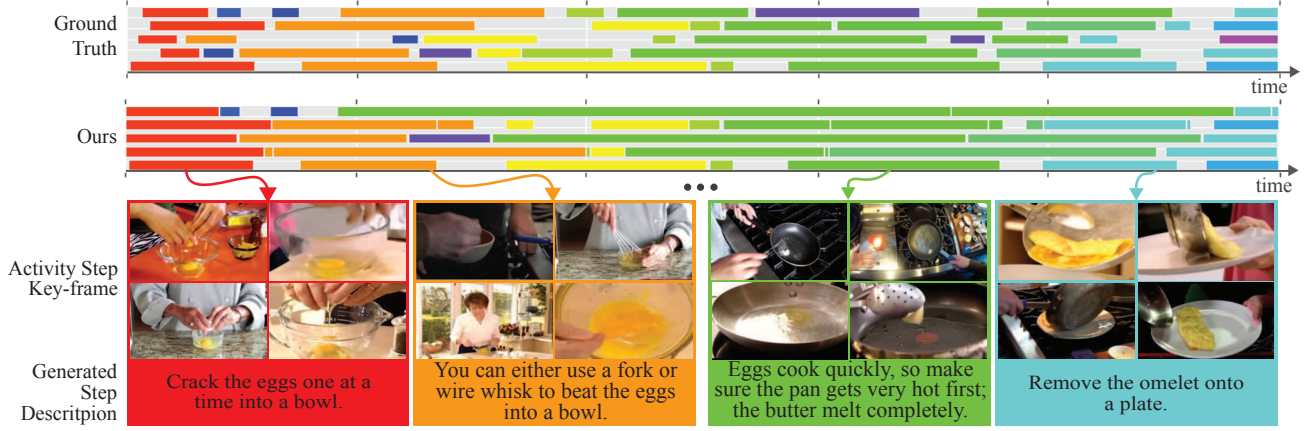
(a) How to make an omelet?
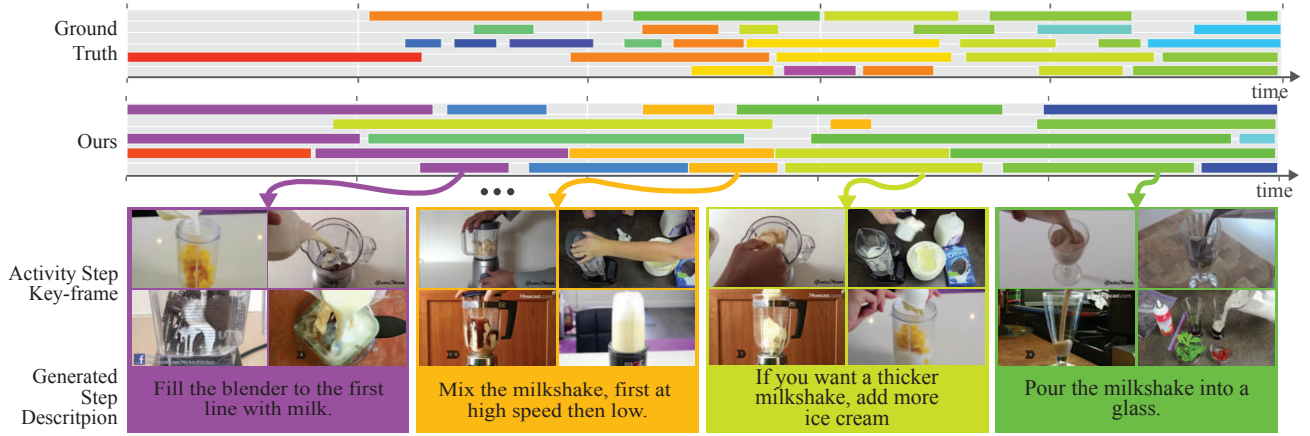
(b) How to make a milkshake?

Figure 8: Temporal segmentation of the videos and ground truth segmentation. We also color code the activity steps we discovered and visualize their key-frames and the automatically generated captions. *Best viewed in color.*
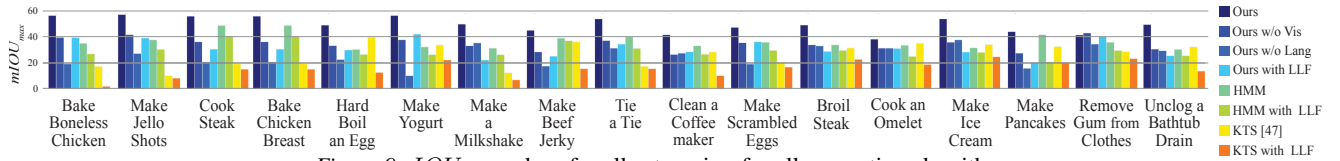

Figure 9: IOU_{cms} values for all categories, for all competing algorithms.
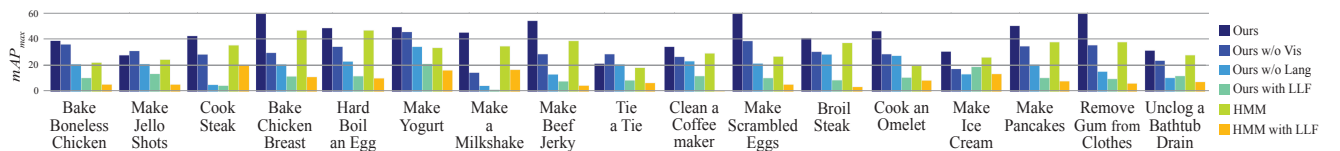

Figure 10: AP_{cms} values for all categories, for all competing algorithms.

use any metric in unsupervised setting. It chooses a matching of ground truth labels with predicted labels by searching over matchings and choosing the ones giving highest score. Therefore, mAP_{csm} and IOU_{csm} are our final metrics.

Accuracy of the temporal parsing. We compute, and plot in Figure 9, the IOU_{cms} values for all competing algorithms and all categories. We also average over the categories and

summarize the results in the Table 1. As the Figure 9 and Table 1 suggest, proposed method consistently outperforms the competing algorithms and its variations. One interesting observation is the importance of both modalities reflected in the dramatic difference between the accuracy of our method and its single modal versions.

Moreover, the difference between our method and HMM

Figure 11: Qualitative results for parsing 'Travel San Francisco' category.

Table 1: Average IOU_{cms} and mAP_{cms} over all categories.

	KTS [48] w/ LLF	KTS[48] w/ Sem	HMM w/ LLF	HMM w/Sem	Ours w/ LLF	Ours w/o Vis	Ours w/o Lang	Our full
IOU_{cms}	16.80	28.01	30.84	37.69	33.16	36.50	29.91	52.36
mAP_{cms}	n/a	n/a	9.35	32.30	11.33	30.50	19.50	44.09

Table 2: Semantic mean-average-precision mAP_{sem}.

	HMM w/ LLF	HMM w/Sem	Ours w/ LLF	Ours w/o Vis	Ours w/o Lang	Our full
mAP_{sem}	6.44	24.83	7.28	28.93	14.83	39.01

is also significant. We believe this is due to the ill-posed definition of activities in HMM since the granularity of the activity steps is subjective. In contrast, our method starts with the well-defined definition of finding set of steps which generate the entire collection. Hence, our algorithm do not suffer from granularity problem.

Coherency and accuracy of activity step discovery. Although IOU_{cms} successfully measures the accuracy of the temporal segmentation, it can not measure the quality of discovered activities. In other words, we also need to evaluate the consistency of the activity steps detected over multiple videos. For this, we use unsupervised version of mean average precision mAP_{cms}. We plot the mAP_{cms} values per category in Figure 10 and their average over categories in Table 1. As the Figure 10 and the Table 1 suggests, our proposed method outperforms all competing algorithms. One interesting observation is the significant difference between semantic and low-level features. Hence, our mid-level features play a key role in linking videos.

Semantics of activity steps. In order to evaluate the role of semantics, we performed a subjective analysis. We concatenated the activity step labels in the grount-truth into a label collection. Then, we ask non-expert users to choose a label for each discovered activity for each algorithm. In other words, we replaced the maximization step with subjective labels. We designed our experiments in a way that each clip received annotations from 5 different users. We randomized the ordering of videos and algorithms during the subjective evaluation. Using the labels provided by subjects, we compute the mean average precision (mAP_{sem}).

Both mAP_{cms} and mAP_{sem} metrics suggest that our method consistently outperforms the competing ones. There is only one recipe in which our method is outperformed by our baseline of no visual information. This is mostly because of the specific nature of the recipe *How to*

tie a tie?. In such videos the notion of object is not useful since all videos use a single object -tie-.

The importance of each modality. As shown in Figure 9 and 10, the performance, consistently across all categories, drops when any of the modalities is ignored. Hence, the joint usage is necessary. One interesting observation is the fact that using only language information performed slightly better than using only visual information. We believe this is due to the less intra-class variance in the language modality (*i.e.*, people use same words for same activities). However, it lacks many details(less complete) and is more noisy than visual information. Hence these results validate the complementary nature of language and vision.

Generalization to generic structured videos We experiment the applicability of our method beyond How-To videos by evaluating it on non-How-To categories. In Figure 11, we visualize the results for the videos retrieved using the query "Travel San Francisco". We showed some of the discovered activities as well as the timeline for five randomly selected videos. The resulting clusters follow semantically meaningful activities and landmarks and show the applicability of our method beyond How-To queries. It is interesting to note that Chinatown and Clement St ended up in the same cluster; considering the fact that Clement St is known for its Chinese food, this shows successful utilization of semantic connections.

7. Conclusions

In this paper, we tried to capture the underlying structure of human communication by jointly considering visual and language cues. We experimentally validated that given a large-video collection having subtitles, it is possible to discover activities without any supervision over activities or objects. Experimental evaluation also suggested the available noisy and incomplete information is powerful enough to not only discover activities but also describe them. We also think that the resulting discovered knowledge can be effectively used in many domains like multimedia interfaces and robot knowledge bases [52].

8. Acknowledgements

References

[1] Suuplementary material for the paper. http://cvgl.stanford.edu/watchandlearn/. 6

[2] Wikihow-how to do anything. http://www.wikihow.com. 6

[3] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, et al. Video in sentences out. *arXiv preprint arXiv:1204.2742*, 2012. 2

[4] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 3. 2

[5] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates making pancakes. In *Humanoids*, 2011. 2

[6] 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. 2

[7] M. Bollini, J. Barry, and D. Rus. Bakebot: Baking cookies with the pr2. In *The PR2 Workshop, IROS*, 2011. 2

[8] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. 3

[9] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013. 2

[10] O. Duchenne, I. Laptev, J. Sivic, F. Bash, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 2

[11] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003. 2

[12] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV 2010*. 2010. 2

[13] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In *CVPR*. IEEE, 2013. 2

[14] E. Fox, M. Hughes, E. Sudderth, and M. Jordan. Joint modeling of multiple related time series via the beta process with application to motion capture segmentation. *Annals of Applied Statistics*, 8(3):1281–1313, 2014. 5

[15] F. Grabler, M. Agrawala, W. Li, M. Dontcheva, and T. Igarashi. Generating photo manipulation tutorials by demonstration. *TOG*, 28(3):66, 2009. 2

[16] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. 2005. 5

[17] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009. 2

[18] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *CVPR*, 2011. 2

[19] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013. 2

[20] M. Jain, J. van Gemert, and C. G. Snoek. University of amsterdam at thumos challenge 2014. 2

[21] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. 2014. 2, 6

[22] V. Kantorov and I. Laptev. Efficient feature extraction, encoding and classification for action recognition. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2014*, 2014. 6

[23] A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. *ArXiv e-prints*, Dec. 2014. 2

[24] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013. 2

[25] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014. 2

[26] G. Kim and E. P. Xing. Reconstructing storyline graphs for image recommendation from web community photos. In *CVPR*, 2014. 2

[27] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *ICML*, 2014. 2

[28] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? text-to-image coreference. In *CVPR*, 2014. 2

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 4

[30] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. A large video database for human motion recognition. In *ICCV*, 2011. 2

[31] T. Lan, L. Chen, Z. Deng, G.-T. Zhou, and G. Mori. Learning action primitives for multi-level video event understanding. In *Workshop on Visual Surveillance and Re-Identification*, 2014. 2

[32] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014. 2

[33] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[34] I. Laptev and P. Pérez. Retrieving actions in movies. In *ICCV*, 07. 2

[35] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012. 2

[36] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 3

[37] T. W. Liao. Clustering of time series dataa survey. *Pattern recognition*, 38(11):1857–1874, 2005. 6

[38] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013. 2

[39] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What's Cookin'? Interpreting Cooking Videos using Text, Speech and Vision. *ArXiv e-prints*, Mar. 2015. 2

[40] J. Malmaud, E. J. Wagner, N. Chang, and K. Murphy. Cooking with semantics. *ACL*, 2014. 2

[41] T. S. Motwani and R. J. Mooney. Improving video activity recognition using object recognition and text mining. In *ECAI*, 2012. 2

[42] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 2

[43] E. Olson, M. Walter, S. J. Teller, and J. J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In *RSS*, 05. 4, 6

[44] D. Oneata, J. Verbeek, and C. Schmid. The lear submission at thumos 2014. 2014. 2

[45] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 2

[46] P. Perona and W. Freeman. A factorization approach to grouping. In *ECCV*. 1998. 4

[47] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014. 2

[48] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*. 2014. 2, 6, 8

[49] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *PROCEEDINGS OF THE IEEE*, pages 257–286, 1989. 6

[50] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for tv baseball programs. In *ACM MM*, 2000. 2

[51] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009. 2

[52] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula. Robo brain: Large-scale knowledge engine for robots. *Tech Report*, Aug 2014. 8

[53] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001. 6

[54] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*, pages 966–973, 2010. 2

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

[56] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 2

[57] C. Sun and R. Nevatia. Discover: Discovering important segments for classification of video events and recounting. In *CVPR*, 2014. 2

[58] M. Tenorth, D. Nyga, and M. Beetz. Understanding and executing instructions for everyday manipulation tasks from the world wide web. In *ICRA*, 2010. 2

[59] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM TOMM*, 3(1):3, 2007. 2

[60] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 2

[61] H. Yu and J. M. Siskind. Grounded language learning from video described with sentences. In *ACL*, 2013. 2

[62] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013. 2

[63] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the visual interpretation of sentences. In *CVPR*, 2013. 2