Unsupervised Semantic Parsing of Video Collections

Anonymous ICCV submission

Paper ID 623

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 and video segments. 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 understand human activities.

Language and vision often provide different, but correlating and complementary information. Challenge lies in that both video frames and language (from subtitles[1]) are only a noisy, partial observation of the actions being performed. However, the complementary nature of language and vision gives the opportunity to robustly understand the activities only from these partial observations. In this paper, we present a unified model, considering both of the modalities, in order to parse human activities into activity steps with no form of supervision.

[1]generated either via Automatic Speech Recognition (ASR) or by the user, are now available for most YouTube videos.

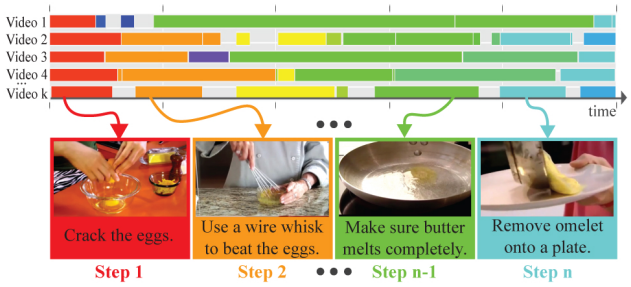


Figure 1: Given a large video collection, we discover semantic activity steps without any supervision by using available multi-modal information (visual frames and subtitles). We also parse each video based on the discovered steps.

The key idea in our approach is the observation that the largecollection of videos, pertaining to the same activity 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 panckage", "How to tie a bow tie"). Instructional videos 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 videos as long as they are composed of a set of steps.

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

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 sub-

titles. 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 using learned atoms. In our method, we do neither use 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 of the activity steps based on the language atoms it frequently uses.

This work is the first to discover activity steps for a complex video collection with no supervision over activions and/or objects. We are also the first to approach this problem in a multimodal (joint language and vision) manner. In addition, our method is capable of providing a caption describing the steps; our approach to captioning is fundamentally different from the majority of existing video/image-to-text work in two aspects: 1) the captions are generated without any supervised caption-clip pairs, 2) our captions are *descriptions* of the semantic steps, yet they are inferred from *narrations*. This is different from the existing captioning work as their reference data is also descriptive of the visual information, while the narration over videos often provides complementary information to the visuals and is not necessarily descriptive.

2. Related Work

Three key aspects differentiate this work from the majority of existing techniques for similar tasks: 1) capability of providing a semantic parsing of a video category leading to a compact storyline representation, 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 [56] and mostly focus on *choosing keyframes* for visualization.

Summarizing videos is particularly important for some specific domains like ego-centric videos and news reports as they are generally long in duration. There are many successful works [34, 36, 49]; however, they mostly rely on characteristics specific to the domain.

Summarization is also applied to the large image collections by recovering the temporal ordering and visual similarity of images [26], and by Gputa et al. [16] to videos in a supervised framework using annotations of actions. The image collections are also used to choose important view points for key-frame selection by Khosla et al.[24] and further extended to video clip selection [25, 47]. 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 [3] in this area focus on learning a common multimodal space in order to jointly represent language and vision. They are further extended to learning higher level relations between object segments and words [51]. Similarly, Zitnick et al.[60, 59] used abstracted clip-arts to understand spatial relations of objects and their language correspondences. Kong et al. [28] and Fidler et al. [12] both accomplished the task of learning spatial reasoning by only using the image captions. Relations extracted from image-caption pairs, are further used to help semantic parsing [58] and activity recognition [40]. Recent works also focus on automatic generation of image captions with underlying ideas ranging from finding similar images and transferring their captions [44] to learning language models conditioned on the image features [27, 52, 11]; their employed approach to learning language models is typically either based on graphical models [11] or neural networks [52, 27, 23].

All aforementioned methods are using 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 human activity recognition is broad. The closest techniques to our problem are either supervised or focus on detecting a particular (and often short) action in a weakly or unsupervised manner. Also, a large body of action recognition methods are intended for trimmed videos clips or remain limited to detecting very short atomic actions [29, 53, 41, 32, 10, 50]. Even though some promising recent works attempted action recognition in untrimmed videos [22, 43, 21], they are primarily fully supervised.

Additionally, several method for localizing instances of actions in rather longer video sequences have been developed [9, 18, 33, 5, 46]. 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 [41, 57, 20, 31, 30] 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 [54, 8, 2]. 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 [38, 55] focus on semantic parsing of language recipes in order to extract actions and the objects in the form of predicates. Tenorth et al.[55] 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 [4, 6] 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.[37] consider both language and vision modalities and propose a method to align an input video to a recipe. However, it 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. There is also an approach which generates multi-modal recipes from expert demonstrations [14]. However, it is developed only for the domain of "teaching user interfaces" and are not applicable 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 **??**). These atoms are designed in a way to be 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 find 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 **??**); Given the series of multi-modal frame representations, we discover set of clusters occurring over multiple videos using a non-parametric Bayesian method (Section **??**). 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.

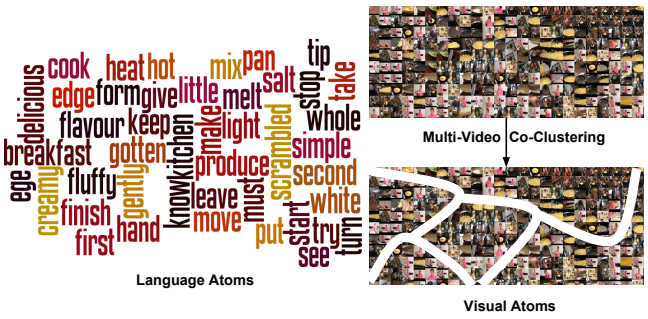


Figure 2: We learn language and visual atoms in order to represent multi-modal information. Language atoms are frequent words and visual atoms are the clusters of object proposals.

Learning Visual Atoms: In order to learn visual atoms, we create a large collection of object proposals by independently generating object proposals from each frame of each video. These proposals are generated using the Constrained Parametric Min-Cut (CPMC) [7] algorithm based on both appearance and motion cues. We note the k^{th} proposal of t^{th} frame of i^{th} 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 [35]) can be applied for this problem, the joint processing of a large video collection requires handling large visual variability among multiple video. We propose a new method to jointly cluster object proposals from multiple videos in Section 5. After the clustering stage, 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 as $D = \bigcup_{i \in N_C} \bigcup_{t \in T^{(i)}} L_t^i$. 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 all words with their "tf-idf" values and choose the top K words as language atoms (*We set* $K = 100$ *in our experiments*). As an example, we show the language atoms

extracted for the activity category *How to make a 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 t^{th} frame of the i^{th} 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 k^{th} 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.

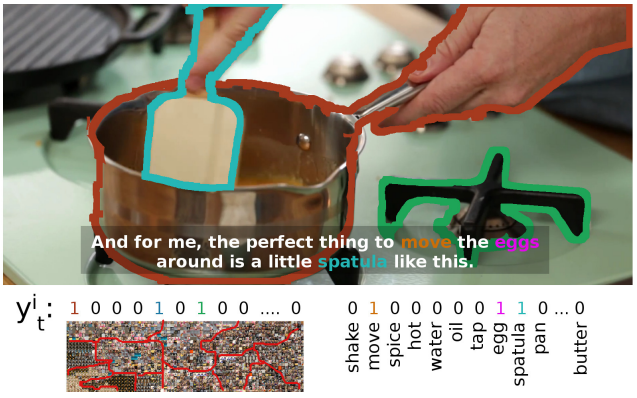


Figure 3: **Visualization of the representation for a sample frame.** Three of the object proposals of the frame is included in the visual atoms and three of the words in the subtitle of the frame is included in the language atoms.

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.

Basic Graph Clustering: Consider the set of object proposals extracted from a single video $\{r_t^k\}$, and a pairwise similarity metric $d(\cdot,\cdot)$ among them. We follow the single cluster graph partitioning (SCGP)[42] approach to find the dominant cluster which maximizes the inter-cluster similarity:

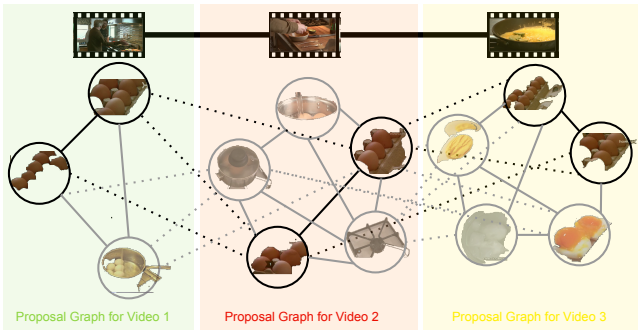


Figure 4: **Joint proposal clustering.** Here, we show the $1^{st}NN$ video graph and $2^{nd}NN$ region graph. Each region proposal is linked to its two NNs from the video it belongs and two NNs from the videos it is neighbour of. Black represents the proposals selected as part of the cluster and gray represents the ones which are not selected. Moreover, dashed lines are intra-video edges and solid ones are inter-video edges.

ity:

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

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 $\arg\max_{\mathbf{x}} \frac{\mathbf{x^T} A \mathbf{x}}{\mathbf{x^T x}}$ This can be solved by finding the dominant eigenvector of $\mathbf{x}$ after relaxing x_t^k to $[0, 1]$ [42, 45]. Upon finding the dominant cluster, the remaining clusters can be found by removing the members of the selected cluster from the collection, and re-applying the same algorithm.

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[**?**]. This hierarchical graph structure is visualized in Figure 4 for 3 videos sample. 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 intra-video relations.

Given the pairwise distance matrices $\mathbf{A^{(i)}}$, the binary indicator vectors $\mathbf{x^{(i)}}$ for each video, and the pairwise distance matrices for video pairs 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 eigen-decomposition based approach from [42, 45] as a result of the modification, we can use Stochastic Gradient Descent (SGD) 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 $r^{(i,j)} = \frac{\mathbf{x^{(i)}}^T \mathbf{A^{(i,j)}} \mathbf{x^{(j)}}}{\mathbf{x^{(i)}}^T \mathbb{1}\mathbb{1}^T \mathbf{x^{(j)}}}$

After finding the cluster by optimizing the cost function, we remove the object proposals belonging to the cluster and re-apply the same algorithm to find the next cluster. After finding $K = 20$ clusters, we discard the remaining object proposals as they were deemed not relevant to the activity. Please note that 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.

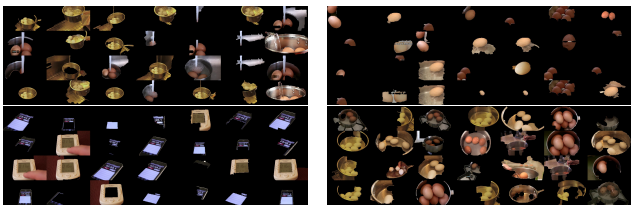


Figure 5: Randomly selected images of four randomly selected clusters learned for *How to hard boil an egg?*

5.1. Unsupervised Activity Representation

In this section, we explain the generative model which we use in order 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 t^{th} frame of the i^{th} video as $z_t^{(i)}$. Since our model is non-parametric, the number of activities are not fixed.

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 now 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.[13] proposed the Beta Process Hidden Markov Models (BP-HMM). In BP-HMM setting, each time-series data exhibits a subset of available features with different ordering. Similarly, in our setup each video exhibits a subset of activity steps.

Our model follows the construction of Fox et al.[13] and differs in the choice of probability distributions since [13] 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;

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

where B_0 and the b_k are determined by the underlying Poisson process [15] 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)[15]. 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 controls the likelihood of the features getting shared by multiple videos.

The above IBP construction represents the activity step discovery part of our method. In addition to the activity step discovery, we also need to model the video parsing over discovered steps. Moreover, we need to model this two steps 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 [13]. 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 by 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 up 1. This procedure can be represented formally as

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

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 all activity steps in the form of a vector. This model is also presented as a graphical model in Figure 6

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.[13]. Our sampling procedure composed of iterative sampling 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 η,π,$\theta_\mathbf{k}$ sampler from the selected activities $\mathbf{f^{(i)}}$. Intuitively, we iterate over discovering activity steps given the temporal activity labels and estimating activity labels given the discovered activities. We give the full details of this sampler in supplementary material.

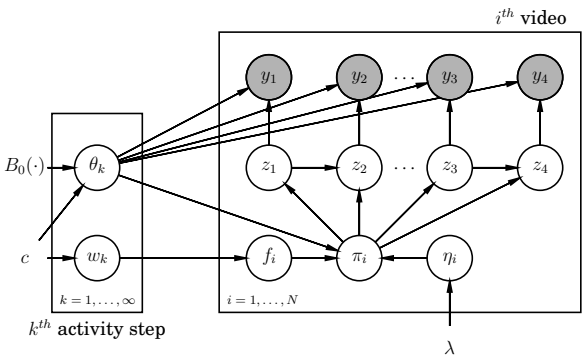


Figure 6: **Graphical model for BP-HMM:** The left plate represent the set of activity steps and the right plate represent the set of videos. Intuitively, the left plate is for activity step discovery and right plate is for video parsing. Given a set of selected steps, the marginal model of the video becomes an Hidden Markov Model. *See Section 5.1.1 for details.*

6. Experiments

In order to experiment the proposed method, we first collected a dataset (details in Section 6.1). We labelled small part of the dataset with frame-wise activity step labels and used the resulting set as an evaluation corpus. Neither the set of labels, nor the temporal boundaries are exposed to our algorithm since the set-up 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 initiate our data collection using WikiHow as obtaining the top100 queries the crowd is interested and choosing the ones which are directly related to the physical world and objects. In other words, we ignore the queries like *How to get over a break up?* as they have no objective set of steps. Resulting queries are;

How to*Bake Boneless Skinless Chicken, Make Jello Shots, Cook Steak, Bake Chicken Breast, Hard Boil an Egg, Make Yogurt, Make a Milkshake, Make Beef Jerky, Tie a Tie, Clean a Coffee Maker, 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 evaluation set, we randomly choose 5 videos out of 100 per query. Hence, we have total of 125 evaluation videos and 2375 unlabelled videos. We label the start and end frames of activity steps (*i.e.* based on the steps of the

recipe) as well as the name of the step. We will release the code and collected dataset at http://anonymous.

6.1.1 Outlier Detection

Since we do not have any expert intervention in our data collection, the resulting collection might have outliers. Main reason for the outliers are the 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 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 large cluster of instructional videos. Given a large video collection, we use the graph we explain in Section 5 and compute the dominant video cluster by using the Single Cluster Graph Partitioning [42] and discards the remaining videos as outlier.

In Figure 7, we visualize some of the discarded videos for various queries. Although our algorithm have false positives while detecting outliers, we always have enough number of videos (minimum 50) after the outlier detection thanks to the large-scale dataset.

Figure 7: **Sample videos which our algorithm discards as an outlier for various queries.** Detected outliers are a toy milkshake, a milkshake charm, a funny video about How to NOT make smoothie, an informative video about the danger of a fire, a cartoon about pancake, a neck-tie video erroneously labeled as bow-tie, a song including the phrase *How to tell if a gold is real?* and a lamb cooking mislabeled as *How to bake chicken?*

6.2. Qualitative Results

After independently running our algorithm on all categories, we discover activity steps and parse the videos according to these discovered steps. We visualize some of these categories qualitatively in Figure 8a and 8b. We show the temporal parsing of 5 evaluation videos as well as 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[**?**] by using the subtitles. Moreover, we generate a caption for each activity step by sampling this model conditioned on the θ_k^l. We explain the details of this process in supplementary material.

As shown in the Figures 8a&8b, resulting steps are semantically meaningful. Moreover, the language captions are also quite informative hence we can conclude that there is enough language context within the subtitles in order to detect activities. On the other hand, some of the activity steps always occur together and our algorithm merges them into a single step while promoting sparsity.

6.3. Quantitative Results

We compare our algorithm with the following baselines in the following sections. **Low-level features (LLF):** In order to experiment the effect of language and visual atom learning, we compare with low-level features in different algorithms. As a feature, we use HOG, HOF and MBH features and use frame-wise Fisher vector representation using the implementation of Oneata et al.[**?**]. As an observation model, we follow the Drichelet distribution since the Fisher vector is a histogram.

Single modality: To experiment the importance of the effect of multi-modal approach, we also compare with single modality based approach whenever possible. We simply experiment based on a single modality.

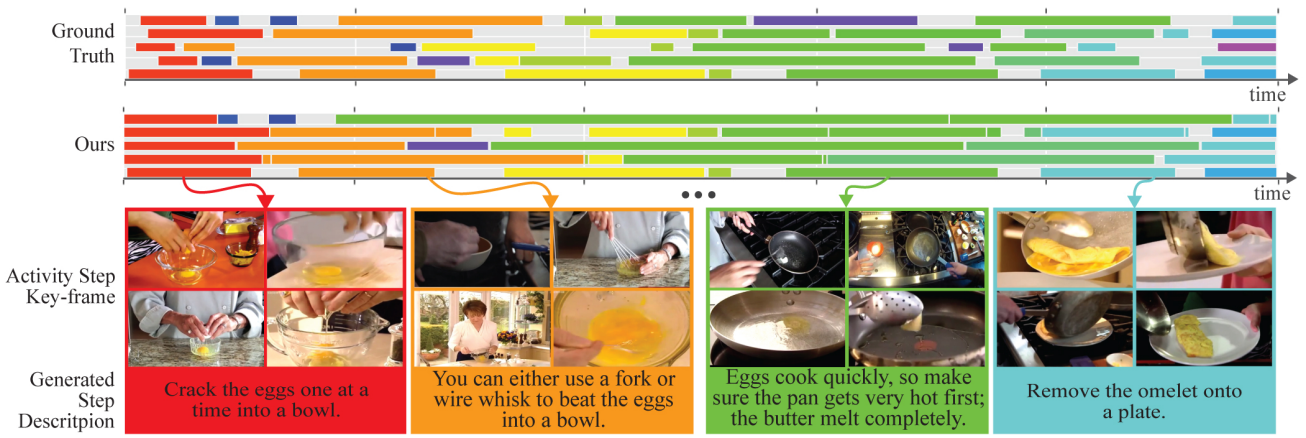
Hidden Markov Model (HMM): To experiment the effect of joint generaive model, we compare our algorithm with an HMM. We use the Baum-Welch algorithm[48] and choose the number of clusters via cross-validation.

Kernel Temporal Segmentation[47]: Kernel Temporal Segmentation (KTS) proposed by Potapov et al.[47] 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.

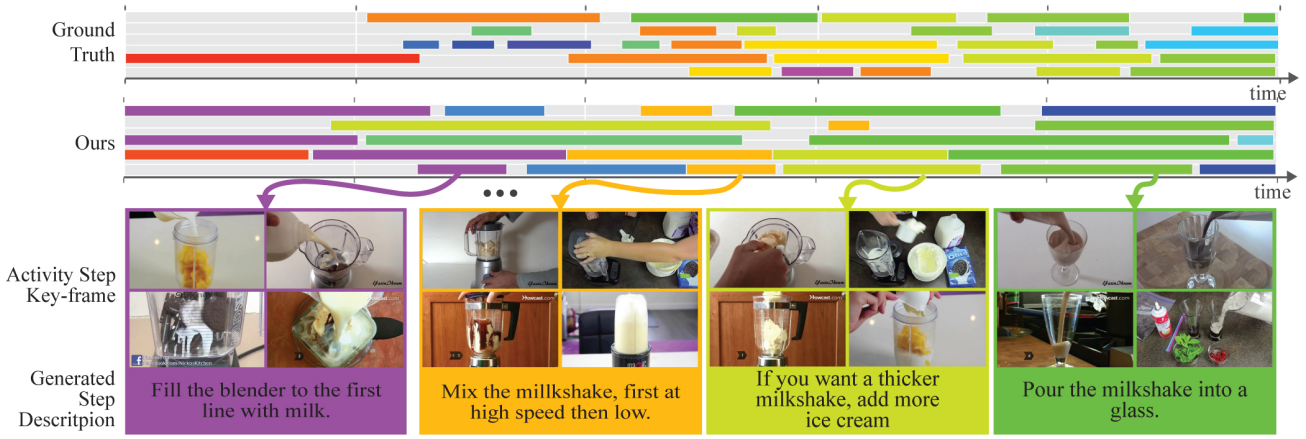
6.3.1 Results

Given parsing results and ground truth, we evaluate quality of both of the temporal segmentation and the activity step discovery.

We base our evaluation on two widely used metrics; intersection over union and mean average precision. Intersection over union measures the quality of temporal segmentation and it is defined as; $IOU = \frac{1}{N}\sum_{i=1}^{N}\frac{\tau_i^\star \cap \tau_i'}{\tau_i^\star \cup \tau_i'}$ where N is the number of segments, $\tau_i^\star$ is ground truth temporal segment and τ_i' is the detected segment. Mean average precision is defined per activity part class and can be computed based on a precision-recall curve following TREC[**?**] definition. In order to adopt this metric into unsupervised setting, we used cluster similarity measure(csm)[**?**] which simply enables us using any metric in unsupervised setting. For any metric, it searches over all possible matching between ground truth label and predicted label and choose the matching which gives the best result. Hence, we are using mAP_{csm} and IOU_{csm} as evaluation metrics.



(a) How to make an omelet?



(b) How to make a milkshake?

Figure 8: Temporal segmentation of the videos by our method and ground truth segmentation. We also color code the discovered activity steps and visualize the key-frames and the automatically generated captions for some of them. *Best viewed in color.*

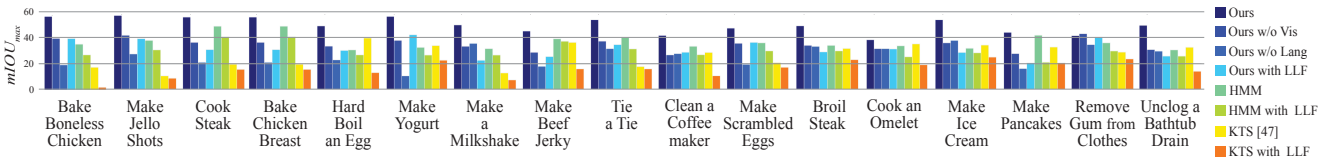


Figure 9: IOU_{max} values for all recipes, for all competing algorithms.

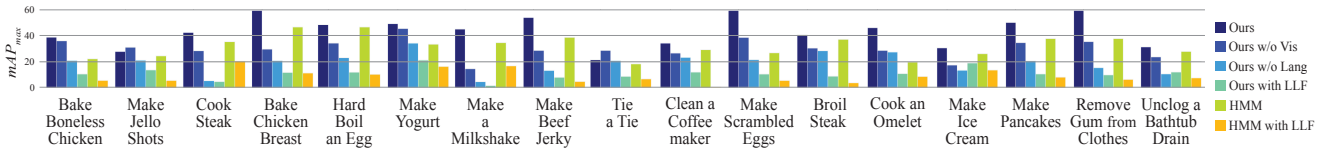


Figure 10: AP_{max} values for all recipes, for all competing algorithms.

Accuracy of the temporal parsing. IOU_{cms} captures the accuracy of the temporal segmentation and we plot the IOU_{cms} values for all competing algorithms for all categories. We also average over the categories and summarize the results in the Table 1. As the Figure 9 and Table 1 suggests, proposed method consistently outperforms the com-

Table 1: Average of IOU_{cms} and mAP_{cms} over recipes.

	KTS [47] w/ LLF	KTS[47] 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

peting algorithms and its variations. One interesting observation is the importance of both modalities as a result of dramatic difference between the accuracy of our method and its single modal versions.

Moreover, the difference between our method and HMM 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. On the other hand, 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, the mid-level features are key for linking multiple videos.

Semantics of activity steps. In order to further 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. By using the activity labels provided by subjects, we compute the mean average precision wrt ground truth call it 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 based line 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 over the entire video. This single object is a *tie* and does not fit the assumption of a frame based on multiple visual atoms.

The importance of each modality. In order to experiment the importance of joint usage of language and vision modalities, we compare our method with a self-baseline of using a single modality. As shown in Figure 9 and 10, our method significantly outperforms both of the baselines consistently in all recipes. 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 more noisy than visual information. Hence these results validate the complementary nature of language and vision.

References

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

[2] 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

[3] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003. 2

[4] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates making pancakes. In *Humanoids*, 2011. 3

[5] 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

[6] M. Bollini, J. Barry, and D. Rus. Bakebot: Baking cookies with the pr2. In *The PR2 Workshop, IROS*, 2011. 3

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

[8] 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

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

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

[11] 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

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

[13] 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, 6

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

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

[16] 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

[17] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*. 2014.

[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] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-S. Chua. Beyond search: event-driven summarization for web videos. *ACM TOMM*, 7(4):35, 2011.

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

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

[22] 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. http://crcv.ucf.edu/THUMOS14/, 2014. 2

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

[30] 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

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

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

[33] I. Laptev and P. Pérez. Retrieving actions in movies. In *ICCV*, 2007. 2

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

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

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

[37] 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. 3

[38] J. Malmaud, E. J. Wagner, N. Chang, and K. Murphy. Cooking with semantics. *ACL*, 2014. 3

[39] S. Mori, H. Maeta, Y. Yamakata, and T. Sasada. Flow graph corpus from recipe texts. In *LREC*, 2014.

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

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

[42] E. Olson, M. Walter, S. J. Teller, and J. J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In *RSS*, 2005. 4, 5, 7

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

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

[45] P. Perona and W. Freeman. A factorization approach to grouping. In *ECCV*. 1998. 4, 5

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

[47] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*. 2014. 2, 7, 9

[48] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *PROCEEDINGS OF THE IEEE*, pages 257–286, 1989. 7

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

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

[51] 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

[52] 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

[53] 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

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

[55] M. Tenorth, D. Nyga, and M. Beetz. Understanding and executing instructions for everyday manipulation tasks from the world wide web. In *ICRA*, 2010. 3

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

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

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

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

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