

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

Youtube2Story: Unsupervised Joint-Representation of Instructional Videos

Anonymous ICCV submission

Paper ID ****

Abstract

The ABSTRACT

1. Introduction

Leaning the instructions of a novel non-trivial task is both a challenge and a necessity for both humans and autonomous systems. This necessity resulted in many community generated instruction collections [?, ?] and expert curated recipe books[?, ?]. However, these instructions are generally based on a language modality and explains a single way of performing the task although there are variety of ways. On the other hand, online video storage services are full of unstructured instructional videos¹ covering variety of ways, environment conditions and view angles. Although there have been many successful attempts in detecting activities from videos [?, ?], structural representation of such a large and useful video collection is not possible. In this paper, we focus on joint semantic representation of YouTube videos as a response to a single query. We specifically study the unsupervised joint-detection of the activities from a collection of YouTube videos.

Understanding of the instructional videos, requires the careful processing of two complementary modalities namely language and the vision. Luckily the target domain, YouTube videos, has unstructured subtitles as well. They are either generated by the content developer (5% of the time) or automatically generated by using the Automatic Speech Recognition (ASR) softwares. The main limitations of the existing activity detection literature for this problem is scalability and representation level. Existing approaches are mainly supervised and requires extensive training set which is not tractable in the scale of YouTube videos. Moreover, current activity detection research focuses on the low-level visual features. However, such videos in the wild have objects with completely different texture and shape characteristics from wide range of views. Instead, we focus on extracting high-level visual semantic representations and us-

ing salient words occurring among the videos.

We rely on the assumption that the videos collected as the response of a same instructional query, share similar activities performed by the similar objects. We start with the independent processing of the videos in order to create a large collection of visual object proposals and words. After the proposal generation, we jointly process the proposal collections and words to detect the visual objects and words which can be used to represent the unstructured information. Since we rely on high-level information instead of the low-level features, the resulting objects represent the semantic information instead of visual characteristic. By using the extracted objects, we compute the holistic representation of the multi-modal information in each frame.

Moving from frame-wise visual understanding to activity understanding, requires the joint processing of all the videos with the temporal information. In order to exploit the temporal information, we model each video as a Hidden Markov Model using state space of activities. Since we assume that the videos share some of the activities and we have no supervision, we use a model based on *beta process mixture model*. Our model jointly learn the activities and detect them in the videos. Moreover, it does not require prior knowledge over the number of activities.

2. Related Work

Video Summarization Summarizing an input video as a sequence of keyframes (static) or as a sequence of video clips (dynamic) is useful for both multi-media search interfaces and retrieval purposes. Early works in the area are summarized in [34] and mostly focus on choosing keyframes for visualization. Idea of choosing key-frames is also extended by using the video tags by Hong et al[11] and using the spatio-temporal information by Gygli et al.[10].

Summarizing videos is crucial for ego-centric videos since the ego-centric videos are generally long in duration. There are many works which successfully segment such videos into a sequence of important shots [19, 21]; however, they mostly rely on specific features of ego-centric videos. Rui et al. [30] proposed another dynamic summarization method based on the excitement in the speech of the

¹YouTube has 281.000 videos for "How to tie a bow tie"

108 reporter. Due to their domain specific designs, these algorithms are not applicable to the general instructional videos.
109

110 Same idea is also applied to the large image collections by recovering the temporal ordering and visual similarity
111 of images [16]. This image collections are further used to choose important view points for video key-frame selection
112 by Khosla et al.[14]. And further extended to video clip selection by Kim et al[15] and Potapov et al.[29]. Although
113 they are different from our approach since they do not use any high-level semantic information or the language information,
114 we experimentally compare our method with them.
115
116

117 **Understanding Multi-Modal Information:** Learning
118 the relationship between the visual and language data is a crucial problem due to its immense multimedia applications.
119 Early methods [1] in the area focus on learning a common multi-modal space in order to jointly represent
120 language and vision. They are also extended to learning higher level relations between object segments and words
121 [31]. Zitnick et al.[37, 36] used abstracted clip-arts to further understand spatial relations of objects and their language correspondences. Kong et al. [18] and Fidler et
122 al. [6] both accomplished the same task by using the image captions only. Relations extracted from image-caption
123 pairs, are further used to help semantic parsing [35] and activity recognition [25]. Recent works also focused on generating
124 image captions automatically using the input image. These methods range from finding similar images and using
125 their captions [27] to learning language modal conditioned on the image [17, 32, 5]. And the methods to learn language
126 models vary from graphical models [5] to neural networks [32, 17, 13].
127

128 All aforementioned methods are using supervised labels either as strong image-word pairs or weak image-caption
129 pairs. On the other hand, our method is fully unsupervised.
130

131 Activity Detection/Recognition:

132 **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 [23, 33] focus on semantic parsing of language recipes in order to extract actions and the objects in the form of predicates. Tenorth et al.[33] further process the predicates in order to form a complete logic plan. Mori et al.[24] also learns the relations of the actions in terms of a flow graph with the help of a supervision. The aforementioned approaches focus only on the language modality and they are not applicable to the videos. We have also seen recent advances [2, 3] in robotics which uses the parsed recipe in order to perform cooking tasks. They use supervised object detectors and report a successful autonomous cooking experiment. In addition to the lan-

133 guage based approaches, Malmaud et al.[22] 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/actions 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 constructing the recipe. There is also an approach which generates multi-modal recipes from expert demonstrations [8]. However, it is developed only for the domain of *teaching user interfaces* and are not applicable to the videos.
134
135

136 3. Method

137 In this section, we explain the high-level components of our method which we visualize in Figure 1. Our proposed method consists of three major components; **(1) Online query and filtering:** Our system starts with querying the YouTube with an *How to* question, and records the top 100 resulting videos. In order to detect the similarity of the videos quickly, we also process the text descriptions of the returned videos, and we represent them as bag-of-words. We further use these representations in order to create a video graph and also to eliminate outliers. **(2) Frame-wise multi-modal representation:** In order to semantically represent the spatio-temporal information in the videos, we process both the visual and language content of each video. We extract the region proposals and jointly cluster them to detect semantic visual objects. For the language descriptions, we detect the salient words of the corpus generated by the concatenation of the subtitles. We finally represent the each frame in terms of the resulting objects and salient words. **(3) Unsupervised joint clustering:** After describing the each frame by using the salient objects and words, we apply a non-parametric Bayesian method in order to find the temporally consistent clusters (collection of video clips) occurring over multiple videos. We expect these clusters to correspond to the fine-grained activities which construct the recipes/high level activities. Moreover, our empirical results suggest that the resulting clusters significantly correlates with the fine-grained activities.
138
139

140 We now explain the details of the each sub-system in the following sections.
141

142 3.1. Video Collection and Outlier Detection

143 As we explain in the Section 3, our system starts with querying the YouTube for the recipe which we want to learn its fine-grained actions. Although we explain how do we choose such queries in Section ?? detail, any query starting with *How to* can be considered as an example. We collect the top 100 videos with their (automatically generated) captions. Youtube generates these captions by using an automatic speech recognition (ASR) algorithm. After obtaining the corpus, we link similar videos to each other by creat-

324

3.2.1 Learning Language Atoms

After obtaining the videos and subtitles belonging to a single query, we concatenate all subtitles into a single collection(term corpus). As a document, we use all the words extracted from all subtitles of all queries. Moreover, we compute the tf-idf as $tfidf(w, d, D) = f_{w,d} \times \log \frac{N}{n_w}$ where w is the word, d is the collection of words corresponding to the query, $f_{w,d}$ is the frequency of the word in the collection d , N is the total number of videos returned from all queries and n_w is the number of videos whose subtitle include the word w . After computing the tf-idf, we sort all words with their tf-idf values and choose the top K words as set of salient words (*We set $K = 100$ in our experiments*).

We show below the top 50 salient words extracted for the query *How to hard boil an egg?*. Moreover, they correspond to important objects, actions and adjectives which can semantically relate actions over multiple videos.

sort, place, water, egg, bottom, fresh, pot, crack, cold, cover, time, over-cooking, hot, shell, stove, turn, cook, boil, break, pinch, salt, peel, lid, point, haigh, rules, perfectly, hard, smell, fast, soft, chill, ice, bowl, remove, aside, store, set, temperature, coagulates, yolk, drain, swirl, shake, white, roll, handle, surface, flat

342
343
344
345
346
347
348
349

3.2.2 Learning Visual Atoms

In order to learn visual atoms, we create a large collection of proposals by independently generating region proposals from each frame of the each video. These proposals are generated by using the Constrained Parametric Min-Cut (CPMC) [4] algorithm by using both appearance and motion cues. We note the k^{th} region of t^{th} frame of i^{th} video as $r_t^{(i),k}$. Moreover, we drop the video index (i) if it is clear from the context.

We follow the spectral graph clustering approach in order to group these regions into semantically meaningful objects similar to the Keysegments approach [20]. However, idea of clustering region proposals into set of semantic objects have been mostly utilized for clusters generated by a single video and they fail to cluster objects having a large visual difference. Hence, we extend this work to spectral joint clustering of region proposals over multiple videos.

367

Joint Proposal Clustering to Detect Visual Objects

Since our proposals are generated from multiple videos, combining them into a single region collection and clustering it is not desired for two reasons; (1) objects have large visual differences among videos and accurately clustering them into a single cluster is hard, (2) clusters are desired to have region proposals from multiple videos in order to semantically relate videos. We propose a joint version of the spectral region clustering algorithm to satisfy these requirements.

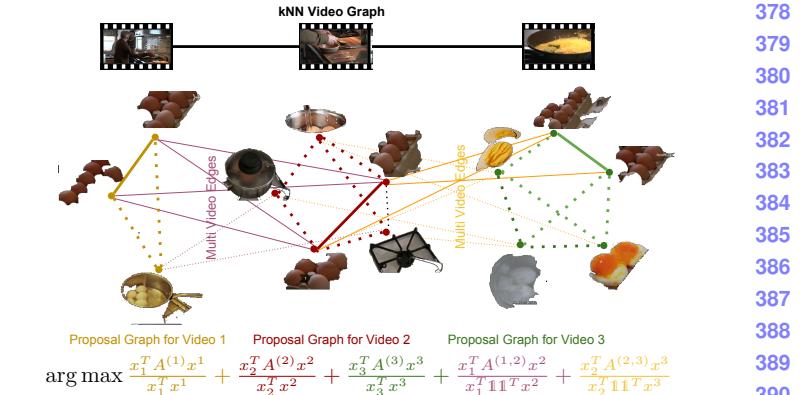


Figure 3: **Visualization of the joint proposal clustering.** Here, we show the 1NN video graph and 2NN region graph. Each region proposal is linked to its 2 nearest neighbours from the video it belongs and 2 nearest neighbours from the videos it is neighbour of.

We first explain the original spectral graph clustering algorithm and then extend it to joint clustering. Consider the set of region proposals extracted from a single video r_t^k , and a similarity metric $d(\cdot, \cdot)$ between any region proposal pair. We follow the single cluster graph partitioning (SCGP)[26] approach to find the dominant cluster which maximizes the inter-cluster similarity. In other words, we solve

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

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. When we use the vector form of the indicator variables as $\mathbf{x}_{tK+k} = x_t^k$ and the pairwise distance matrix as $\mathbf{A}_{t_1 K + k_1, t_2 K + k_2} = d(r_{t_1}^{k_1}, r_{t_2}^{k_2})$, this equation can be compactly written as $\arg \max_{\mathbf{x}} \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$. Moreover, it can be solved by finding the dominant eigenvector of \mathbf{x} after relaxing x_t^k to $[0, 1]$ [26, 28]. After finding the maximum, the remaining clusters can be found by removing the selected region proposals from the collection, and re-applying the same algorithm for the second dominant cluster.

Our extension of the SCGP into multiple videos is based on the assumption that the important objects of recipes occur in most of the videos. Hence, we re-formulate the problem by relating videos to each other. We use the kNN graph of the videos which we used for the outlier detection as explained in the Section ???. Moreover, we also create the kNN graph of region proposals in each video. This hierarchical graph structure is also visualized in Figure 3 for 3 videos. After creating this graph, we impose the similarity of regions in the selected cluster coming from each video as well as the similarity of regions coming from neighbour

432 videos. Hence, given the pairwise distance matrices $\mathbf{A}^{(i)}$,
 433 binary indicator vectors $\mathbf{x}^{(i)}$ for each video and pairwise
 434 distance matrices for video pairs as $\mathbf{A}^{(i,j)}$, we define our
 435 optimization problem as;

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

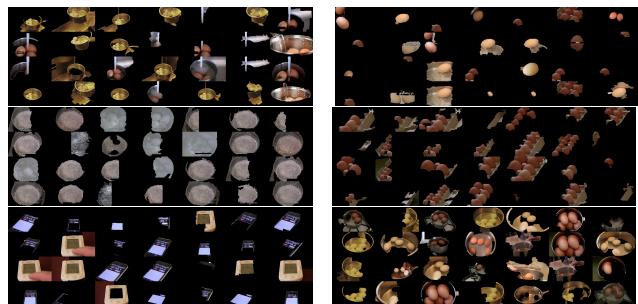
440 where $\mathcal{N}(i)$ is the neighbours of the video i in the kNN
 441 graph, $\mathbf{1}$ is vector of ones and N is the number of videos.
 442 We visualize this optimization objective in Figure 3 for the
 443 case of 3 videos.

444 After changing the optimization function, we can not use
 445 the efficient eigen-decomposition based approach from [26,
 446 28]; however, we can use stochastic gradient descent (SGD)
 447 since the cost function is quasi-convex when it is relaxed.
 448 We use the SGD with the following gradient function;

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

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

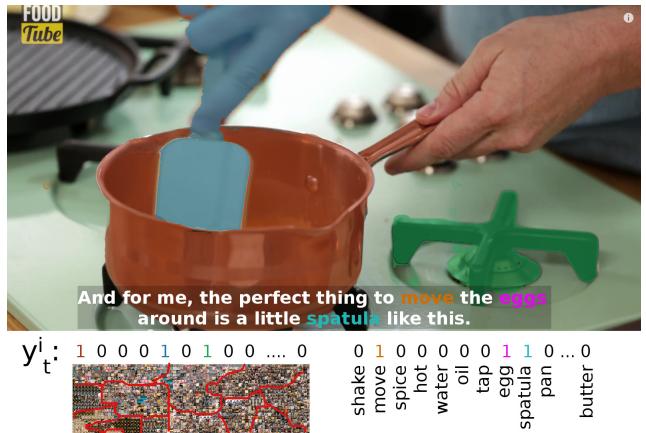
450 After finding the dominant cluster by optimizing the cost
 451 function, we remove the selected cluster and re-aply the
 452 same algorithm to find the next dominant cluster. After
 453 finding $K = 20$ clusters, we discard the remaining region
 454 proposals. In Figure 4, we visualize some of the clusters
 455 which our algorithm generated after applied on the videos
 456 returned by the query *How to Hard Boil an Egg*. As shown
 457 the figure, the resulting clusters are highly correlated and
 458 correspond to semantic objects&concepts.



465 Figure 4: Randomly selected images of randomly selected
 466 clusters learned for *How to hard boil an egg*?
 467
 468
 469
 470
 471
 472
 473
 474

480 3.2.3 Multi-Modal Representation of Frames

482 After learning the objects and salient words, we represent
 483 each frame via the occurence of salient words and objects.
 484 Formally, representation of the t^{th} frame of the i^{th} video
 485 is denoted as $\mathbf{y}_t^{(i)}$ and computed as $\mathbf{y}_t^{(i)} = [\mathbf{y}_t^{(i),1}, \mathbf{y}_t^{(i),v}]$



486 Figure 5: **Visualization of the representation of a sample**
 487 **frame.** 3 of the region proposals of the frame is included in
 488 the object clusters and 3 of the words in the subtitle of the
 489 frame is included in the salient word list.

490 such that k^{th} entry of the $\mathbf{y}_t^{(i),1}$ is 1 if the subtitle of the
 491 frame has the k^{th} word and 0 otherwise. $\mathbf{y}_t^{(i),v}$ is also a
 492 binary vector similarly defined over objects. We visualize
 493 the representation of a sample state in the Figure 5.

500 3.3. Unsupervised Activity Representation

501 In this section, we explain the generative model which
 502 we use in order to jointly learn the activities from videos.
 503 We start with explaining the notation. As we already de-
 504 fined in the previous sections, we note the extracted frame
 505 representation of the frame t of video i as $\mathbf{y}_t^{(i)}$. Moreover,
 506 we model our algorithm based on activities and the note the
 507 activity of the t^{th} frame of the i^{th} video as $z_t^{(i)}$. Since our
 508 model is non-parametric, the number of activities are not
 509 fixed i.e. $z_t^{(i)} \in \mathcal{N}$.

510 We model each activity as a Bernoulli distribution over
 511 the visual and language atoms as $\theta_k = [\theta_k^l, \theta_k^v]$ such that
 512 m^{th} entry of the θ_k^l represents the likelihood of seeing
 513 m^{th} language word in the frame having activity k . Simi-
 514 larly, m^{th} entry of the θ_k^v represents the likelihood of
 515 seeing m^{th} object. In other words, each frame's re-
 516 presentation $\mathbf{y}_t^{(i)}$ is sampled from its activity distribution as
 517 $\mathbf{y}_t^{(i)} | z_t^{(i)} = k \sim Ber(\theta_k)$. As a prior over θ , we use its con-
 518 jugate distribution – Beta distribution –.

519 In the following sections, we first explain the generative
 520 model which links activities and frames. Then, we explain
 521 how this model can be jointly learned and inferred by using
 522 the combination of Gibbs sampling and Metropolis-Hastings
 523 samplers.

3.3.1 Beta Process Hidden Markov Model

For joint understanding of the time-series information, Fox et al.[7] proposed the Beta Process Hidden Markov Models (BP-HMM). It relies on the set of features(activities in our case) which can explain the behaviour of all time-series data (all videos in our case). In BP-HMM setting, each time-series data exhibits a subset of available features.

In our model, each video i chooses a set of activities through an activity vector $\mathbf{f}^{(i)}$ such that $f_k^{(i)}$ is 1 if i^{th} video has activity k , it is 0 otherwise. When the activity vectors of all videos are concatenated, it becomes an activity matrix \mathbf{F} such that i^{th} row of the \mathbf{F} is the activity vector $\mathbf{f}^{(i)}$. Moreover, each feature k also has a prior probability b_k and a distribution parameter θ_k . Distribution parameter θ_k is the Bernoulli distribution as explained in the Section 3.3. Moreover, its base distribution (B_0) is the *Beta random variable*. In this setting, the activity 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 [9] and the feature vector is determined as independent Bernoulli draws as $f_k^{(i)} \sim Ber(b_k)$. After marginilizing over the b_k and θ_k , this distribution is shown to be equivalent to Indian Buffet Process [9]. Where videos are customers and activities are dishes in the buffet. The first video chooses a Poisson(γ) unique dishes. The following video i chooses previously sampled activity k with probability $\frac{m_k}{i}$, proportional to the number of videos (m_k) chosen the activity k , and it also chooses Poisson($\frac{\gamma}{i}$) new activities. Here, γ controls the number of selected activities in each video and β controls the likelihood of the features getting shared by multiple videos.

After each video chooses a subset of activities, we model the videos as an Hidden Markov Model (HMM) over the selected activities. Each frame has the hidden state activity $\text{id}(z_t^{(i)})$ and we observe the binary representation $y_t^{(i)}$. Since we model each activity as a Bernoulli distribution, the emmition probabilities follow the Bernoulli distribution as $p(y_t^{(i)} | z_t^{(i)}) = \text{Ber}(\theta_{z_t^{(i)}})$. Following the construction of the Fox et al.[7], we sample the transition probabilities from a normalized Gamma distribution. For each video i , we sample a Gamma random variable for the transition between activity j and activity k if both of the activities 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 have proper transition probabilities. This procedure can be represented formally as

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

Where κ is the persistence parameter promoting the self state transitions to have more coherent temporal boundaries, \circ is the element-wise product and π_j^i is the transition probabilities in video i from state j to all states in the form of a vector. This model is also presented as a graphical model in Figure 6

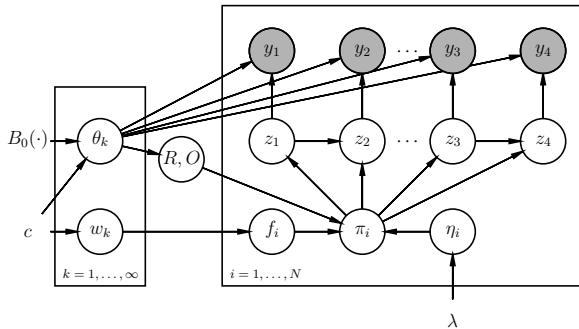


Figure 6: Graphical model for BP-HMM: The left plate represent the set of activities and right plate represent the set of videos. Each video choose a subset of activities through $f^{(i)}$ and transition probabilities between them. After the features are selected, the marginal model of each video becomes an Hidden Markov Model. See the text for the details.

3.3.2 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.[7]. It marginilizes over blah and blah and sample blah and blah. For faster convergence, we also utilize a series of data driven samplers. Here we only discuss the proposed data driven samplers and move the details of the remainin samplers to the Supplementary Material.

Sampling the activity assignments

Sampling the HMM parameters

4. Experiments

In order to experiment the proposed method, we first collected a dataset guided by the human preferences captured via the statistics of a popular online recipe collection –WikiHow [?]. After collecting the dataset, we labelled small part of the dataset with frame-wise activity labels and used the resulting set as an evaluation corpus. Neither the set of labels, nor the temporal boundaries are exposed to the

648 competing algorithm since the set-up is completely unsupervised. We experiment our algorithm against the set of
 649 unsupervised clustering baselines and state-of-the-art algorithms from video summarization literature which are applicable.
 650 In the rest of this section, we first explain the dataset we collected and labelled in detail. Then, we explain the
 651 method which we compare our method against. After explaining the metrics we use, we give both qualitative and
 652 quantitative results. Due to the space limitation, we defer some of the results to the supplementary material.
 653
 654
 655
 656
 657
 658

659 4.1. Dataset

660 We guide our datacollection effort with human preferences based on WikiHow [?] statistics. After obtaining
 661 the top100 queries people interested in WikiHow, we chose
 662 top25 ones which are directly related to the physical world
 663 and objects. We ignore the queries like *How to get over a*
 664 *break up?* and *How to write a resignation Letter?*. Resulting
 665 25 queries are;
 666

667 **How toBake Boneless Skinless Chicken, Cook Steak in a Frying Pan,**
 668 **Make Jello Shots, Tell if Gold Is Real, Bake Chicken Breast, Hard Boil an**
 669 **Egg, Make Pancakes, Tie a Bow Tie Broil Steak, Make a Grilled Cheese**
 670 **Sandwich, Make Scrambled Eggs, Tie a Tie, Clean a Coffee Maker, Make**
 671 **a Milkshake, Make Yogurt, Unclog a Bathtub Drain, Cook an Omelette,**
 672 **Make a Smoothie, Poach an Egg, Cook Lobster Tails, Make Beef Jerky,**
 673 **Remove Gum from Clothes, Cook Ribs in the Oven, Make Ice Cream, Tell**
 674 **if an Egg is Bad**

675 For each of the recipe, we queried YouTube and crawled
 676 the top 100 videos. We also downloaded the english subtitles
 677 if they exist. For evaluation set, we choose 5 videos out
 678 of 100 per query. Hence, we have total of 125 evaluation
 679 videos and 2375 unlabelled videos. We label the start and
 680 end frames of fine-grained activities (*i.e.* steps of the recipe)
 681 as well as their labels. We also release the collect dataset at
 682 <http://anonymous.edu/MMRecipe>.

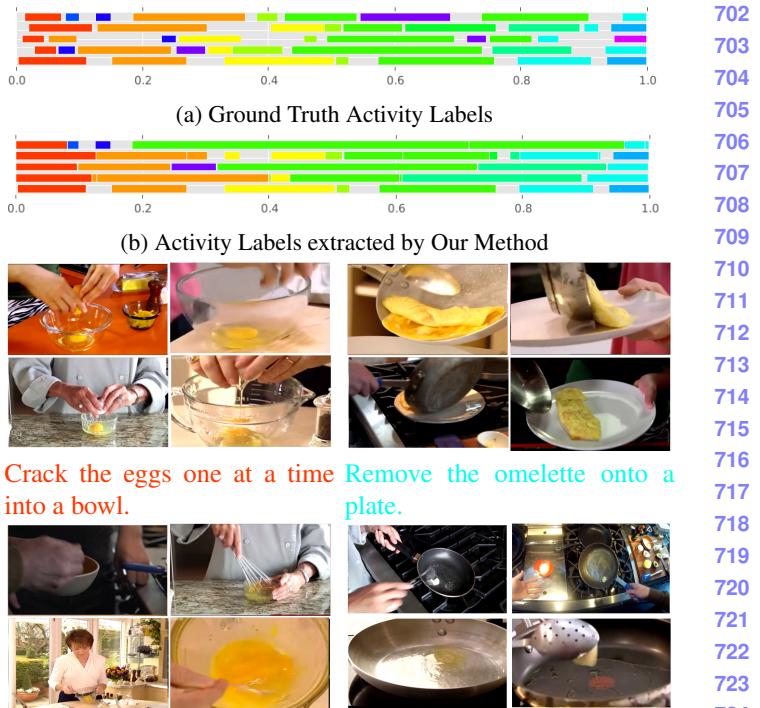
684 4.2. Implementation Details

685 Parameters:

688 **Aligning Clusters:** While comparing the results of our al-
 689 gorithm with the ground truth, we have an alignment prob-
 690 lem. Our algorithm generates arbitrary IDs for clusters and
 691 the cluster IDs are not necessarily matching the ground truth
 692 IDs since the method is unsupervised. For example, we can
 693 name the activity 1 of ground truth as activity 3 although
 694 their content is same. So, we apply an alignment procedure
 695 and choose the alignment of cluster IDs which maximizes
 696 the intersection over union with the ground truth. We apply
 697 this method to all competing algorithms for fairness.

698 4.3. Qualitative Results

699 In this section, we visualize some of the results of our
 700 recipe understanding method. After running our algorithm
 701



702
 703
 704
 705
 706
 707
 708
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755

You can either use a fork or Eggs cook quickly, so make wire whisk to beat the eggs into sure the pan gets very hot first; the butter melt completely.

728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755

Figure 7: Temporal segmentation of the videos by our method and ground truth segmentation. We also color code the learned activity labels and visualize sample frames and the automatically generated captions for some of them. *Best viewed in color.*

independently on all 25 recipes according to the details we explain in Section 4.2, we obtain set of clusters which correspond to the activities. These clusters have set of objects and words; moreover, we also have video clips from multiple videos corresponding to activities. We visualize some of the recipes qualitatively in Figure 7 and ???. We show the temporal segmentation of 5 evaluation videos as well as the segmentation we compute. Moreover, we also color code the clusters to visualize how well the semantic activities are learned.

To visualize the content of each cluster, we display informative frames from different videos. We also train a 3rd order Markov language model[?] by using the subtitles covered by the cluster. Moreover, we generate a caption for each cluster by sampling this model conditioned on the θ_k^l . We explain the details of this process in supplementary material since it is orthogonal to the algorithm and only included for qualitative analysis of language information.

As shown in the Figures 7&???, resulting clusters are semantically meaningful and correspond to the real activities.

756 Moreover, the language captions are also quite informative
 757 hence we can conclude that there is enough language context
 758 within the subtitles in order to detect activities. On the other hand,
 759 some of the activities in the ground truth are not detected by our algorithm and they got merged into
 760 other clusters because they generally occur only in a very few videos.
 761
 762
 763

764 4.4. Quantitative Results

765 4.4.1 Baselines

766 We compare our algorithm with the following baselines in
 767 the following sections.
 768
 769

770 **HMM with semantic features:** In order to experiment
 771 the importance of joint processing of the videos, we compare
 772 our algorithm with independent temporally coherent
 773 clustering of each video. We are using Hidden Markov
 774 Models with Baum-Welch algorithm[?] as a clustering
 775 method and choose the number of clusters with cross-
 776 validation.
 777
 778

779 **BP-HMM with low-level features:** In order to experiment
 780 the importance of defining objects, we also train our
 781 algorithm without using extracted object. We simply
 782 temporally oversegment the video and represent each segment
 783 by using state of the art low-level features from the activity
 784 detection literature [?]. We are using dense trajectory
 785 features for this purpose.
 786
 787

788 **Category specific summary[29]:** An algorithm proposed
 789 by Potapov et al.[29] can detect the temporal boundaries
 790 of the events/activities in the video from a time series data
 791 without any supervision. It enforces a local similarity of
 792 each resultant segment.
 793
 794

795 4.4.2 Metrics

796 **Maximum Intersection over Union (IOU_{max}):** In order
 797 to evaluate the accuracy of the temporal segmentation
 798 of the activities, we use intersection-over-union(IOU).
 799 For N ground truth temporal activity segments (τ_i^* , $i \in N$), N' computed segments (τ'_i , $i \in N'$) and matching
 800 function $m(\cdot)$ such that i^{th} ground truth segment is
 801 matched to $m(i)^{th}$ computed segment, we define IOU as
 802 $IOU = \frac{1}{N} \sum_{i=1}^N \frac{\tau_i^* \cap \tau'_{m(i)}}{\tau_i^* \cup \tau'_{m(i)}}$. Since the matching function is
 803 unknown in the supervised setting, we use the maximum
 804 intersection-over-union while doing exhaustive search over
 805 all matchings as; $IOU_{max} = \max_{m(\cdot)} \frac{1}{N} \sum_{i=1}^N \frac{\tau_i^* \cap \tau'_{m(i)}}{\tau_i^* \cup \tau'_{m(i)}}$
 806
 807
 808
 809

810 **Maximum Average Precision (AP_{max}):** Since the
 811 IOU_{max} is computed per video, it does not capture the
 812 accuracy of the detected activities over multiple videos.
 813 Hence, we also evaluate maximum average precision.
 814 Given matching function $m(\cdot)$, the average precision is defined as $\frac{1}{N} \sum_{i=1}^N \frac{tp_i}{tp_i + fp_i}$ where tp_i is the number of frames correctly labeled with activity i and fp is the number of frames falsely labelled as activity i . Note that this metric is defined over all the videos in the recipe and can only be high if the same activities from multiple videos clustered into a single activity. Similarly, the maximum is over matching functions as $AP_{max} = \max_{m(\cdot)} \frac{1}{N} \sum_{i=1}^N \frac{tp_i}{tp_i + fp_i}$

815 Semantic Correctness:

816 4.4.3 Results

817 Are the activities detected accurately?

818 Are the same activities in different videos linked to each other?

819 Semantics vs Syntax:

820 How important is each modality?

821 Is joint clustering helpful?

822 5. Discussions and Conclusions

823 Discuss which recipes worked and why. Discuss the importance
 824 of semantic representation, scaling features and multi-modality.
 825
 826

827 References

- [1] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003. 2
- [2] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 529–536. IEEE, 2011. 2
- [3] M. Bollini, J. Barry, and D. Rus. Bakebot: Baking cookies with the pr2. In *The PR2 Workshop: Results, Challenges and Lessons Learned in Advancing Robots with a Common Platform, IROS*, 2011. 2
- [4] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010. 4

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

Table 1: Notation of the Paper

$y_t = [y_t^v, y_t^l]$	feature representation of t^{th} frame	I_t	t^{th} frame of the video	$x_{i,r}^p$	1 if p^{th} cluster has r^{th} proposal of i^{th} video
x^p	binary vector for p^{th} cluster	L_t	subtitle for t^{th} frame	$O^{k,k'}$	1 if $\#(z_t = k, z_{t'} = k') = 0 \forall t \leq t'$
$\Theta_k = [\Theta_k^v, \Theta_k^l]$	emmition prob. of k^{th} activity	z_t	activity ID of frame t	f_i^k	1 if i^{th} video has k^{th} activity 0.o.w.
$\eta_i^{k,k'}$	$P(z_{t+1} = k' z_t = k)$ for i^{th} vid	$\pi_i^{k,k'}$	$\eta_i^{k,k'} \times f_i^k \times f_i^{k'}$		

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

- [5] 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 *Computer Vision–ECCV 2010*, pages 15–29. Springer, 2010. [2](#)
- [6] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1995–2002. IEEE, 2013. [2](#)
- [7] 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. [6](#)
- [8] F. Grabler, M. Agrawala, W. Li, M. Dontcheva, and T. Igarashi. Generating photo manipulation tutorials by demonstration. *ACM Transactions on Graphics (TOG)*, 28(3):66, 2009. [2](#)
- [9] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. 2005. [6](#)
- [10] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *Computer Vision–ECCV 2014*, pages 505–520. Springer, 2014. [1](#)
- [11] 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 Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 7(4):35, 2011. [1](#)
- [12] M. C. Hughes and E. B. Sudderth. Nonparametric discovery of activity patterns from video collections. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 25–32. IEEE, 2012.
- [13] A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. *ArXiv e-prints*, Dec. 2014. [2](#)
- [14] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2698–2705. IEEE, 2013. [2](#)
- [15] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 4225–4232. IEEE, 2014. [2](#)
- [16] G. Kim and E. P. Xing. Reconstructing storyline graphs for image recommendation from web community photos. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3882–3889. IEEE, 2014. [2](#)
- [17] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *Proceedings of the 31st Interna-*

- tional Conference on Machine Learning (ICML-14)*, pages 595–603, 2014. [2](#)
- [18] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? text-to-image coreference. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3558–3565. IEEE, 2014. [2](#)
- [19] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, volume 2, page 6, 2012. [1](#)
- [20] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011. [4](#)
- [21] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2714–2721. IEEE, 2013. [1](#)
- [22] 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](#)
- [23] J. Malmaud, E. J. Wagner, N. Chang, and K. Murphy. Cooking with semantics. *ACL 2014*, page 33, 2014. [2](#)
- [24] S. Mori, H. Maeta, Y. Yamakata, and T. Sasada. Flow graph corpus from recipe texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 2370–2377, 2014. [2](#)
- [25] T. S. Motwani and R. J. Mooney. Improving video activity recognition using object recognition and text mining. In *ECAI*, pages 600–605, 2012. [2](#)
- [26] E. Olson, M. Walter, S. J. Teller, and J. J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In *Robotics: Science and Systems*, pages 265–272, 2005. [3](#), [4](#), [5](#)
- [27] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems*, pages 1143–1151, 2011. [2](#)
- [28] P. Perona and W. Freeman. A factorization approach to grouping. In *Computer Vision ECCV’98*, pages 655–670. Springer, 1998. [4](#), [5](#)
- [29] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *Computer Vision–ECCV 2014*, pages 540–555. Springer, 2014. [2](#), [8](#)
- [30] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for tv baseball programs. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 105–115. ACM, 2000. [1](#)

- 972 [31] R. Socher and L. Fei-Fei. Connecting modalities: Semi- 1026
973 supervised segmentation and annotation of images using un- 1027
974 aligned text corpora. In *Computer Vision and Pattern Recog- 1028
975 nition (CVPR), 2010 IEEE Conference on*, pages 966–973. 1029
976 IEEE, 2010. 2 1030
- 977 [32] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. 1031
978 Ng. Grounded compositional semantics for finding and de- 1032
979 scribing images with sentences. *Transactions of the Associa- 1033
980 tion for Computational Linguistics*, 2:207–218, 2014. 2 1034
- 981 [33] M. Tenorth, D. Nyga, and M. Beetz. Understanding and ex- 1035
982 ecuting instructions for everyday manipulation tasks from the 1036
983 world wide web. In *Robotics and Automation (ICRA), 2010 1037
984 IEEE International Conference on*, pages 1486–1491. IEEE, 1038
2010. 2 1039
- 985 [34] B. T. Truong and S. Venkatesh. Video abstraction: A sys- 1040
986 tematic review and classification. *ACM Transactions on 1041
987 Multimedia Computing, Communications, and Applications*, 1042
988 3(1):3, 2007. 1 1043
- 989 [35] H. Yu and J. M. Siskind. Grounded language learning from 1044
990 video described with sentences. In *ACL (1)*, pages 53–63, 1045
991 2013. 2 1046
- 992 [36] C. L. Zitnick and D. Parikh. Bringing semantics into fo- 1047
993 cus using visual abstraction. In *Computer Vision and Pat- 1048
994 tern Recognition (CVPR), 2013 IEEE Conference on*, pages 1049
995 3009–3016. IEEE, 2013. 2 1050
- 996 [37] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning 1051
997 the visual interpretation of sentences. In *Computer Vi- 1052
998 sion (ICCV), 2013 IEEE International Conference on*, pages 1053
999 1681–1688. IEEE, 2013. 2 1054
- 1000 1055
- 1001 1056
- 1002 1057
- 1003 1058
- 1004 1059
- 1005 1060
- 1006 1061
- 1007 1062
- 1008 1063
- 1009 1064
- 1010 1065
- 1011 1066
- 1012 1067
- 1013 1068
- 1014 1069
- 1015 1070
- 1016 1071
- 1017 1072
- 1018 1073
- 1019 1074
- 1020 1075
- 1021 1076
- 1022 1077
- 1023 1078
- 1024 1079
- 1025