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 challange and a necessity for both humans and autonomous systems. This necessity resulted in many community generated instruction collections [**?**, **?**] and expert curated recipe books[**?**, **?**]. However, this 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 videos[1] covering variety of ways, environment conditions and view angles. Although there have been many succesfull 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 specficially 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 modaliities 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 using salient words occuring 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 understading 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 key-frames 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 succesfully segment such videos in to a sequence of important shots [19, 21]; however, they mostly rely on specific features of ege-centric videos. Rui et al. [30] proposed another dynamic summarization method based on the excitement in the speech of the

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

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

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

Understanding Multi-Modal Information: Learning the relationship between the visual and language data is a crucial problem due to its immense multimedia applications. Early methods [1] in the area focus on learning a common multi-modal space in order to jointly represent language and vision. They are also extended to learning higher level relations between object segments and words [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 al. [6] both accomplished the same task by using the image captions only. Relations extracted from image-caption pairs, are further used to help semantic parsing [35] and activity recognition [25]. Recent works also focused on generating image captions automatically using the input image. These methods range from finding similar images and using their captions [27] to learning language modal conditioned on the image [17, 32, 5]. And the methods to learn language models vary from graphical models [5] to neural networks [32, 17, 13].

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

Activity Detection/Recognition:

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 successfull autonomous cooking experiment. In addition to the language 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.

3. Method

In this section, we explain the high-level compenents 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) occuring 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.

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

3.1. Video Collection and Outlier Detection

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-

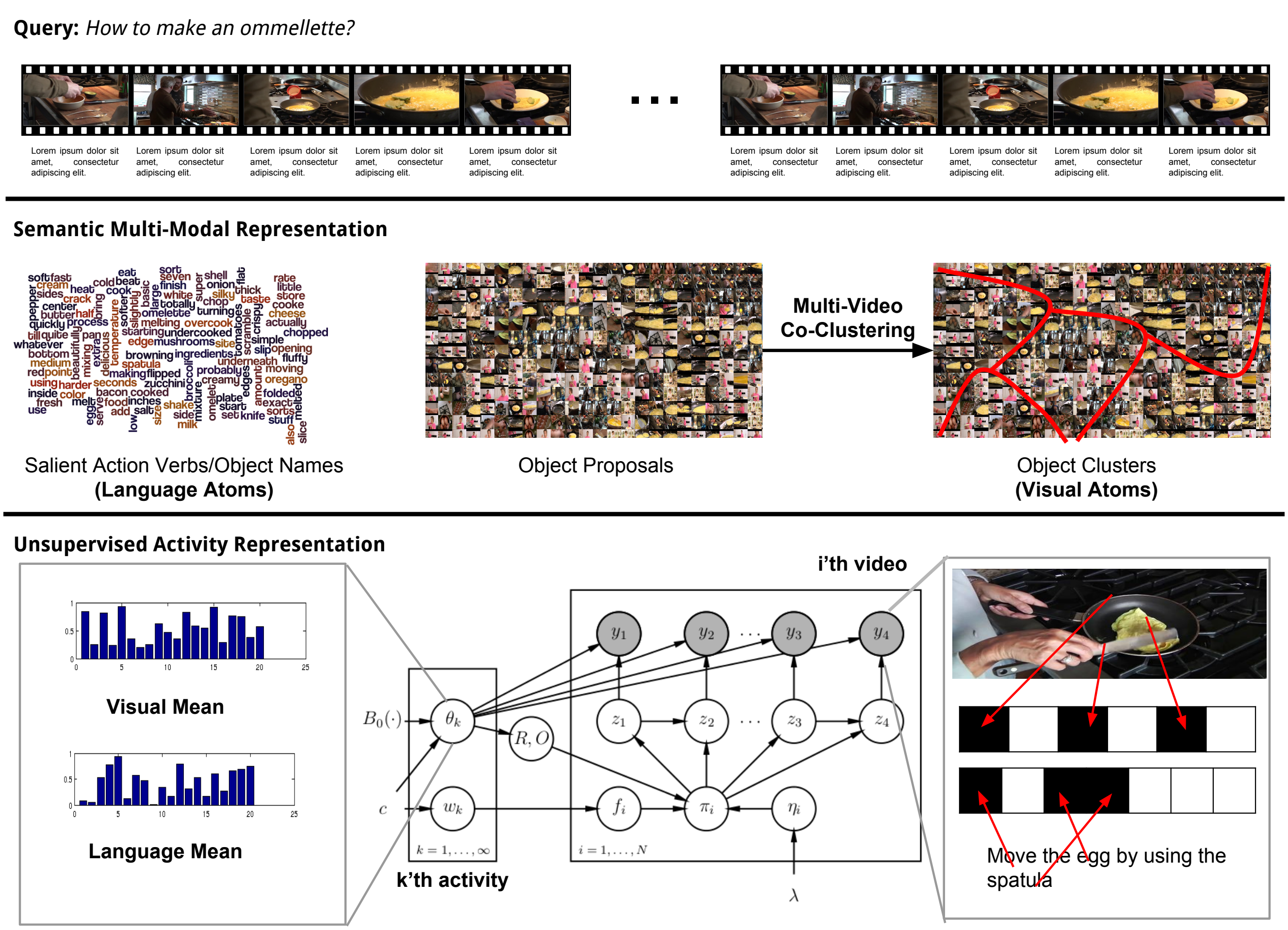


Figure 1: Components of our recipe understanding method. **Query:** We query the YouTube for top 100 *How To* videos and filter the outliers; **Framewise Representation:** We automatically extract object clusters and salient word in order to find multi-modal representation of each frame. **Unsupervised Activity Detection:** We jointly cluster videos in order to learn activities/steps related to the recipe.

ing a kNN video graph. As a distance metric, we use the bag-of-words model of video descriptions. We compute the bag-of-words representation of each description and use χ^2-distance of them as a distance metric. After the creation of the graph, we compute the dominant cluster by using the Single Cluster Graph Partitioning (SCGP)[26] discards the remaining videos as outlier.

As an example, in Figure 2 we visualize some of the discarded videos for the query *How to make a milkshake?*. As shown in the figure, they are actual outliers like making a toy milkshake, making a milkshake charm and a funny video about How to NOT make milkshakes.

3.2. Semantic Multi-Modal Frame Representation

We represent the each frame of the each video by using set of language and visual atoms which we automatically extract. Language atoms are the salient words learned by ranking the words in the subtitle corpus via tf-idf like measure. And, the visual atoms are found by clustering over region proposals which we extract from each frame. We explain the details of proposal generation, clustering and ranking in the subsequent sections.

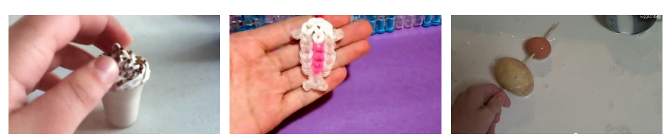


Figure 2: **Sample videos which our algorithm discards as an outlier for the query *How to make a milkshake?*.** These videos are about a toy milkshake, a milkshake charm and a funny video about How to NOT make milkshakes.

3.2.1 Learning Language Atoms

After obtaining the videos and subtitles belonging to a single query, we concatanate 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.

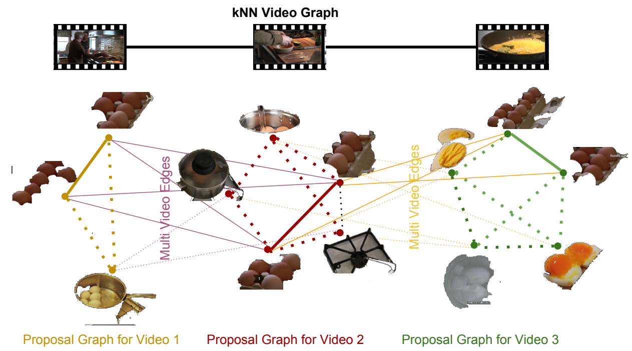
sort, place, water, egg, bottom, fresh, pot, crack, cold, cover, time, overcooking, hot, shell, stove, turn, cook, boil, break, pinch, salt, peel, lid, point, haigh, rules, perfectly, hard, smell, fast, soft, chill, ice, bowl, remove, aside, store, set, temperature, coagulates, yolk, drain, swirl, shake,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

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 apperance 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 utilizied 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.

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 reaons; (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.



$$\arg\max \frac{x_1^T A^{(1)} x^1}{x_1^T x^1} + \frac{x_2^T A^{(2)} x^2}{x_2^T x^2} + \frac{x_3^T A^{(3)} x^3}{x_3^T x^3} + \frac{x_1^T A^{(1,2)} x^2}{x_1^T \mathbb{1}\mathbb{1}^T x^2} + \frac{x_2^T A^{(2,3)} x^3}{x_2^T \mathbb{1}\mathbb{1}^T x^3}$$

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 neihbours 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

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

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. When we use the vector form of the indicator variables as $\mathbf{x_{tK+k}} = x_t^k$ and the pairwise distance matrix as $\mathbf{A}_{t_1K+k_1, t_2K+k_2} = d(r_{t_1}^{k_1}, r_{t_2}^{k_2})$, this equation can be compactly written as $\arg\max_{\mathbf{x}} \frac{\mathbf{x^T A x}}{\mathbf{x^T x}}$ 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-aplying 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 videos. Hence, given the pairwise distance matrices $\mathbf{A}^{(\mathbf{i})}$, binary indicator vectors $\mathbf{x}^{(\mathbf{i})}$ for each video and pairwise distance matrices for video pairs as $\mathbf{A}^{(\mathbf{i},\mathbf{j})}$, we define our optimization problem as;

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

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. We visualize this optimization objective in Figure 3 for the case of 3 videos.

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

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

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

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

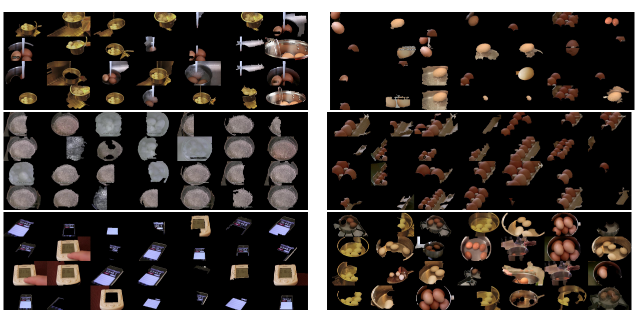


Figure 4: Randomly selected images of randomly selected clusters learned for *How to hard boil an egg?*

3.2.3 Multi-Modal Representation of Frames

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

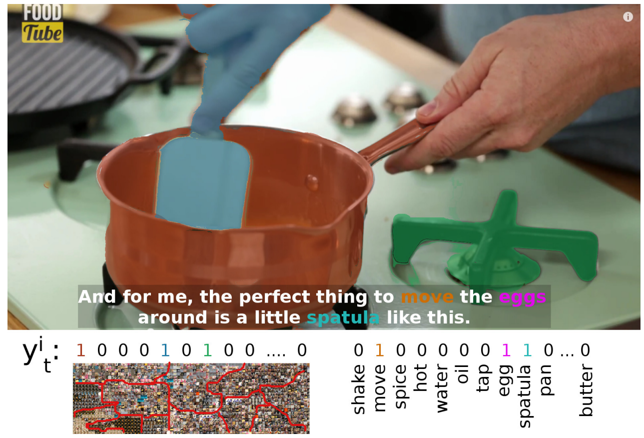


Figure 5: **Visualization of the representation of a sample frame.** 3 of the region proposals of the frame is included in the object clusters and 3 of the words in the subtitle of the frame is included in the salient word list.

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

3.3. Unsupervised Activity Representation

In this section, we explain the generative model which we use in order to jointly learn the activities from videos. We start with explaining the notation. As we already defined in the previous sections, we note the extracted frame representation of the frame t of video i as $\mathbf{y}_{\mathbf{t}}^{(\mathbf{i})}$. Moreover, we model our algorithm based on activities and the note the activity 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 *i.e.* $z_t^{(i)} \in \mathcal{N}$.

We model each activity as 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 represents the likelihood of seeing m^{th} language word in the frame having activity k. Similarly, m^{th} entry of the θ_k^v represents the likelihood of seeing m^{th} object. In other words, each frame's representation $\mathbf{y}_{\mathbf{t}}^{(\mathbf{i})}$ is sampled from its activity distribution as $\mathbf{y}_{\mathbf{t}}^{(\mathbf{i})}|z_t^{(i)} = k \sim Ber(\theta_k)$. As a prior over θ, we use its conjugate distribution – *Beta distribution* –.

In the following sections, we first exlpain the generative model which links activities and frames. Then, we explain how this model can be jointly learned and inferred by using the combination of Gibbs sampling and Metropolis-Hasting 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 $\mathbf{f}^{(\mathbf{i})}$ 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 concatanated, it becomes an activity matrix $\mathbf{F}$ such that i^th row of the $\mathbf{F}$ is the activity vector $\mathbf{f}^{(\mathbf{i})}$. 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 paremeters θ_k and b_k follow the *beta process* as;

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

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 id($z_t^{(i)}$) and we observe the binary representation $\mathbf{y}_\mathbf{t}^{(\mathbf{i})}$. Since we model each activity as a Bernoulli distribution, the emmition probabilities follow the Bernoulli distribution as $p(\mathbf{y}_\mathbf{t}^{(\mathbf{i})}|z_t^{(i)}) = Ber(\theta_{z_t^{(i)}})$. 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

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

Where κ is the persistence parameter promoting the self state transitions to have more coherent temporal bounderies, $\circ$ 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

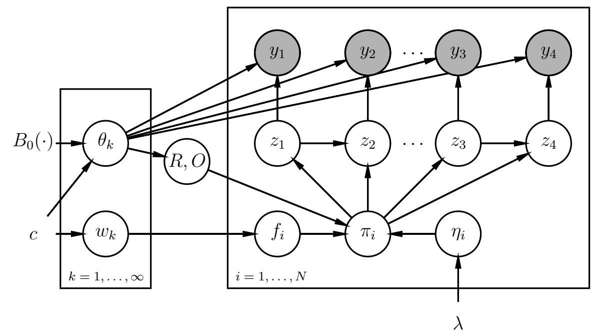


Figure 6: **Graphical model for BP-HMM:** The left plate represent the set of activities and right plate represent the set of videos. Each video choose a subset of activities through $\mathbf{f}^{(\mathbf{i})}$ and transition probabilitiesi between them. After the features are selected, the marginal model of the 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.

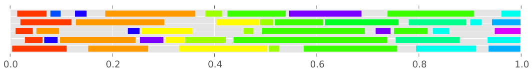
Sampling the activity assignments

Sampling the HMM parameters

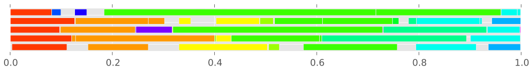
4. Experiments

4.1. Dataset

We collected blah blah videos from YouTube and did blah blah... All numbers etc. These are recipes, 5 of them are evaluation set and labelled temporally and labels are matched.



(a) Ground Truth Activity Labels



(b) Activity Labels extracted by Our Method

(c) Sample images and the automatically generated captions for some of the clusters.

4.2. Baselines

We compare against BP-HMM-O with local feauters and HMM with Semantic Features, HMM with local features, HMM with CNN feautes...

4.3. Qualitative Results

4.4. Metrics

4.5. Accuracy over Activity Detection

4.6. Accuracy over Activity Learning

4.7. Accuracy over Semantic Representation

5. Discussions and Conclusions

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

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

[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

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 $\#(z_t = k, z_{t'}) = k') = 0 \forall \quad t \leq t'$
$\Theta_k = [\Theta_k^v, \Theta_k^l]$	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 $0o.w.$
$\eta_i^{k,k'}$	$P(z_{t+1} = k' \vert z_t = k)$ for i^{th} vid	$\pi_i^{k,k'}$	$\eta_i^{k,k'} \times f_i^k \times f_i^{k'}$		

[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 International 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 Nineth 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 VisionECCV'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

[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

[31] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 966–973. IEEE, 2010. 2

[32] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. 2

[33] M. Tenorth, D. Nyga, and M. Beetz. Understanding and executing instructions for everyday manipulation tasks from the world wide web. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1486–1491. IEEE, 2010. 2

[34] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1):3, 2007. 1

[35] H. Yu and J. M. Siskind. Grounded language learning from video described with sentences. In *ACL (1)*, pages 53–63, 2013. 2

[36] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3009–3016. IEEE, 2013. 2

[37] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the visual interpretation of sentences. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1681–1688. IEEE, 2013. 2