

000
001
002
003
004
005
006
007
008
009
010
011

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 utilizing both visual and language cues in a joint generative model. The proposed method can also provide a textual description for each of 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 new and impactful problem.

1. Introduction

Human communication is highly structured. For instance, in the description an event or a procedure, a clear beginning, end, and certain steps between these two can be usually identified. Parsing such human-generated communication into set of semantic steps is the key to understand human activities. In this paper, we propose a method for decomposing a video into semantically meaningful steps in a fully unsupervised manner.

Language and vision are the two main modalities employed for this structured communication by humans. Explaining “how-to” perform a certain task is a typical example since both verbal (e.g., Do-It-Yourself books) and visual (e.g., instructional YouTube videos¹) ways are both used.

These two modalities often provide different, but correlating and complementary information. We utilize these two modalities (video frames and imperfect automatically generated subtitles) in a unified manner in our framework; we qualitatively and quantitatively argue that a joint inference is crucial for a successful semantic parsing, particularly when no supervision is employed. Note that subtitles,

¹YouTube has 281.000 videos for “How to tie a bow tie”

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

We evaluate the proposed approach on instructional videos from YouTube (e.g., “Making pancake”, “How to tie a bow tie”) as they typically have clear steps and provide concrete grounds for demonstrating a semantically meaningful parsing. These videos are often long and manifest a high intra-class variability yet the underlying steps remains well-defined and structured, similar to almost all human communications. In principle, 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 (see Fig. ??). This storyline provides what particular steps are taking place in the video (*what*), what their semantic meaning is (*how*), and when they are occurring (*when*). This method is also capable of putting multiple videos performing the same overall task in common ground (i.e., the semantic steps space) and capture their high-level similarity, and therefore, provide a *categorical* storyline as well.

In a nutshell, given a video, we capture the visual properties of unsupervised object proposals from each frame as well as the frequencies of keywords from the subtitle. We then employ a generative *beta process mixture model*, which identifies the semantic steps shared among the videos of to the same category based on both text and visual cues. The model also identifies the text keywords which were deemed highly related to the semantic action of each steps. We later learn a language model to provide a textual description of the semantic steps based on the identified keywords.

This work is the first to provide a semantic storyline for a complex video collection. 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 in an unsupervised manner, 2) our captions are *descriptions* of the semantic steps, yet they are inferred from *narration* of the text. This is different from the exist-

108 ing captioning work as their reference data is also descriptive
109 of the visual information, while narration text often provides
110 complementary information to the visuals and is not
111 necessarily descriptive.
112

113 2. Related Work 114

115 Three key aspects, differentiate this work from the
116 majority of existing techniques for similar tasks, are; 1) pro-
117 viding a semantic parsing of a video category leading to
118 a compact storyline representation, 2) being unsupervised,
119 3) adopting a multi-modal joint vision-language model for
120 parsing. A thorough review of the related literature is pro-
121 vided in this Section.
122

123 **Video Summarization** Summarizing an input video as a
124 sequence of key frames (static) or as a sequence of video
125 clips (dynamic) is useful for both multimedia search
126 interfaces and retrieval purposes. Early works in the area
127 are summarized in [35] and mostly focus on choosing
128 keyframes for visualization. Keyframes are also improved
129 by using the video tags by Hong et al.[12] and using the
130 spatio-temporal information by Gygli et al.[11].
131

132 Summarizing videos is particularly important for ego-
133 centric videos as they are generally long in duration. There
134 are many works which successfully segment such videos
135 into a sequence of important shots [19, 21]; however, they
136 mostly rely on the specific characteristics of edge-centric
137 videos. Rui et al. [31] proposed another dynamic sum-
138 marization method based on the excitement of the speech of
139 the reporter. Due to their domain specific designs, these
140 algorithms are neither applicable to the general video col-
141 lections nor to the instructional videos.
142

143 Summarization is also applied to the large image col-
144 lections by recovering the temporal ordering and visual simi-
145 larity of images [16]. This image collections are further
146 used to choose important view points for video key-frame
147 selection by Khosla et al.[14]. And further extended to
148 video clip selection by Kim et al.[15] and Potapov et al.[29].
149 We provide a fresh approach to video summarization by
150 performing it through semantic parsing. Although existing
151 summarization techniques are dissimilar to ours in terms of
152 using high-level semantics or language information, we ex-
153 perimentally compare our method with them.
154

155 **Understanding Multi-Modal Information:** Learning
156 the relationship between the visual and language data is
157 a crucial problem due to its immense applications. Early
158 methods [2] in the area focus on learning a common multi-
159 modal space in order to jointly represent language and vi-
160 sion. They are further extended to learning higher level
161 relations between object segments and words [32]. Simi-
162 larly, Zitnick et al.[38, 37] used abstracted clip-arts to un-
163 derstand spatial relations of objects and their language cor-
164 respondences. Kong et al. [18] and Fidler et al. [7] both
165 accomplished the task of learning spatial reasoning by only
166 using the image captions. Relations extracted from image-
167 caption pairs, are further used to help semantic parsing [36]
168 and activity recognition [25]. Recent works also focused on
169 generating image captions automatically using the input im-
170 age. These methods range from finding similar images and
171 using their captions [27] to learning language modal condi-
172 tioned on the image [17, 33, 6]. And the methods to learn
173 language models vary from graphical models [6] to neural
174 networks [33, 17, 13].
175

176 All aforementioned methods are using supervised labels
177 either as strong image-word pairs or weak image-caption
178 pairs. On the other hand, our method is fully unsupervised.
179

180 Activity Detection/Recognition: 181

182 **Recipe Understanding** Following the interest in commu-
183 nity generated recipes in the web, there have been many
184 attempts to automatically process recipes. Recent meth-
185 ods on natural language processing [23, 34] focus on se-
186 mantic parsing of language recipes in order to extract ac-
187 tions and the objects in the form of predicates. Tenorth
188 et al.[34] further process the predicates in order to form a
189 complete logic plan. Mori et al.[24] also learns the rela-
190 tions of the actions in terms of a flow graph with the help
191 of a supervision. The aforementioned approaches focus
192 only on the language modality and they are not applicable
193 to the videos. The recent advances [3, 4] in robotics use
194 the parsed recipe in order to perform cooking tasks. They
195 use supervised object detectors and report a successful au-
196 tonomous cooking experiment. In addition to the language
197 based approaches, Malmaud et al.[22] consider both lan-
198 guage and vision modalities and propose a method to align
199 an input video to a recipe. However, it can not extract
200 the steps/actions automatically and requires a ground truth
201 recipe to align. On the contrary, our method uses both vi-
202 sual and language modalities and extracts the actions while
203 autonomously constructing the recipe. There is also an ap-
204 proach which generates multi-modal recipes from expert
205 demonstrations [9]. However, it is developed only for the
206 domain of *teaching user interfaces* and are not applicable
207 to the videos.
208

209 3. Method 210

211 In this section, we explain the high-level components
212 of our method which we visualize in Figure 1. Our pro-
213 posed method consists of three major components; **(1) On-**
214 **line query and filtering:** Our system starts with querying
215 the YouTube with an *How to* question, and records the top
216 100 resulting videos. In order to detect the similarity of
217

216

Query: How to make an ommellette?

270

217

271

218

272

219

273

220

274

221

275

222

276

223 **Semantic Multi-Modal Representation**

277

224

278

225

279

226

280

227

281

228

282

229

283

230

284

231

285

232

286

233 **Unsupervised Activity Representation**

287

234

288

235

289

236

290

237

291

238

292

239

293

240

294

241

295

242

296

243

297

244

298

245

299

246

300

247

301

248

302

249

303

250

304

251

305

252

306

253

307

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)



'Lorem ipsum dolor sit amet, consectetur adipiscing elit.'

222

276

223

277

224

278

225

279

226

280

227

281

228

282

229

283

230

284

231

285

232

286

233

287

234

288

235

289

236

290

237

291

238

292

239

293

240

294

241

295

242

296

243

297

244

298

245

299

246

300

247

301

248

302

249

303

250

304

251

305

252

306

253

307

254

308

255

309

256

310

257

311

258

312

259

313

260

314

261

315

262

316

263

317

264

318

265

319

266

320

267

321

268

322

269

323

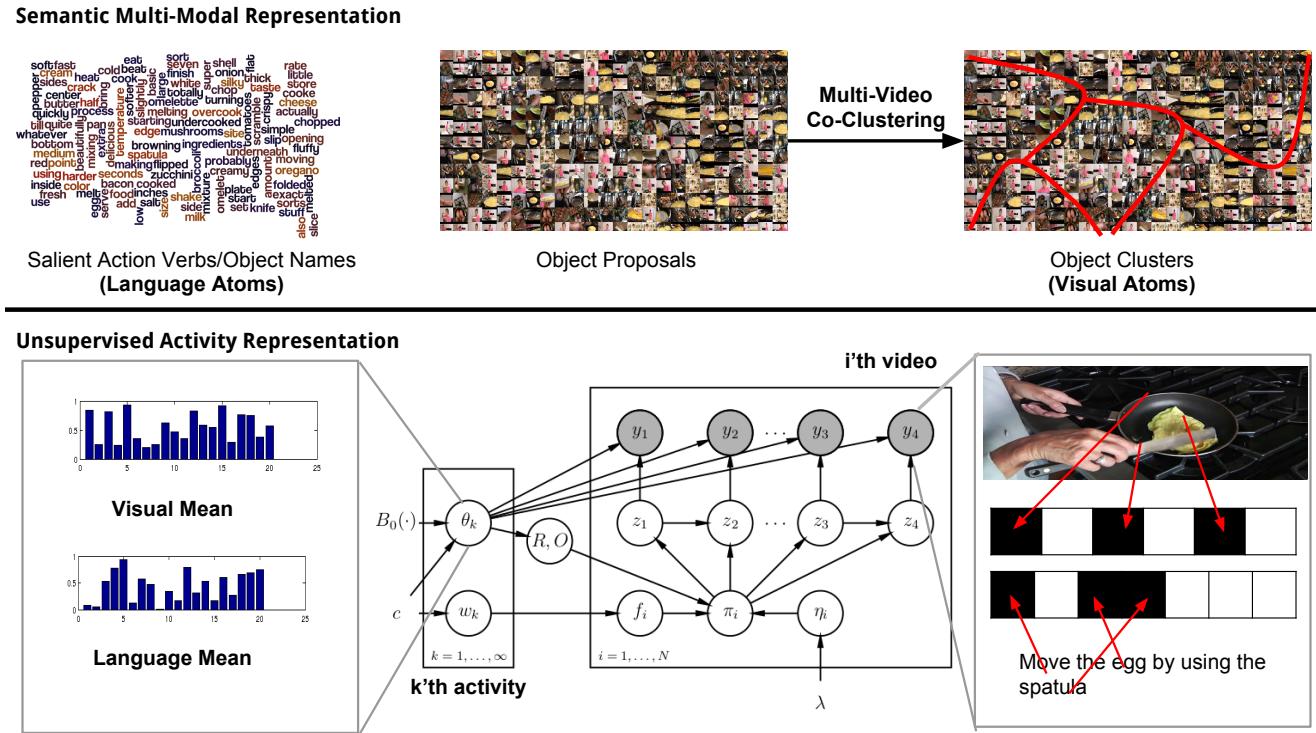
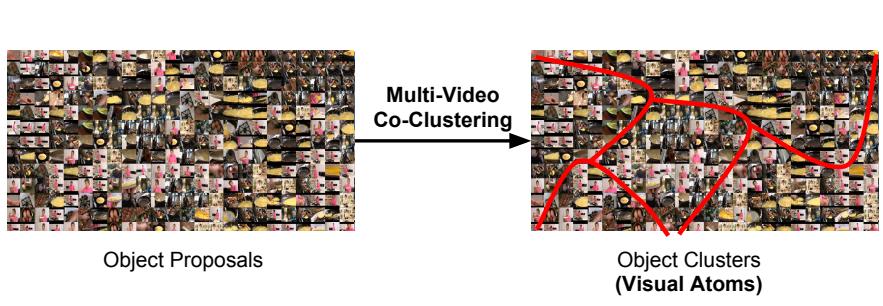


Figure 1: Components of our recipe understanding method. **Query:** We query the YouTube for top 100 *How To* videos and filter the outliers; **Framewise Representation:** We automatically extract object clusters and salient word in order to find multi-modal representation of each frame. **Unsupervised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

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.

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 4.1 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 Au-

324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377

tomatic Speech Recognition (ASR) algorithm. After obtaining the corpus, we link similar videos to each other by creating 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.



399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431

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. 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.

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, high, 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) [5] 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.

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.

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 $\arg \max_{\mathbf{x}} \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$

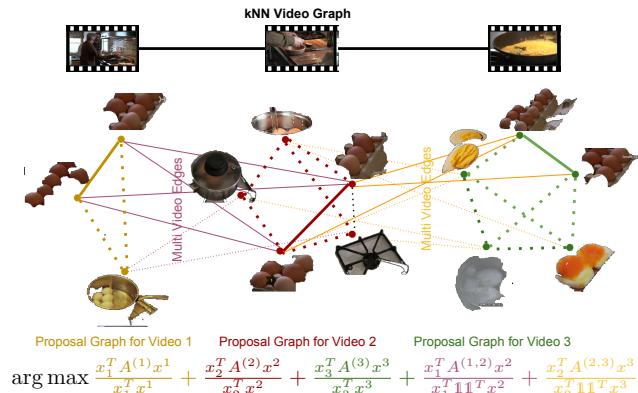


Figure 3: Visualization of the joint proposal clustering. Here, we show the 1NN video graph and 2NN region graph. Each region proposal is linked to its 2 nearest neighbours from the video it belongs and 2 nearest neighbours from the videos it is neighbour of.

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 3.1. 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 $\mathbf{A}^{(i)}$, binary indicator vectors $\mathbf{x}^{(i)}$ for each video and pairwise distance matrices for video pairs as $\mathbf{A}^{(i,j)}$, we define our optimization problem as;

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

where $\mathcal{N}(i)$ is the neighbours of the video i in the kNN graph, $\mathbf{1}$ 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;

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

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

After finding the dominant cluster by optimizing the cost function, we remove the selected cluster and re-apply 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.

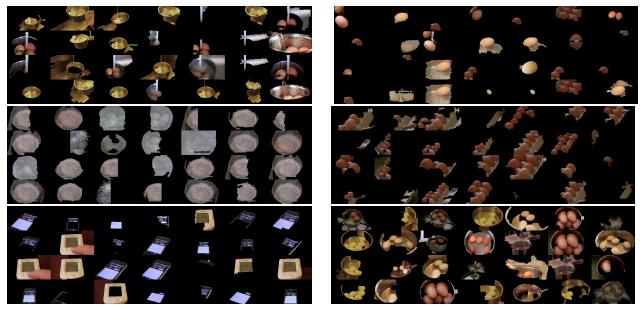


Figure 4: Randomly selected images of randomly selected clusters learned for *How to hard boil an egg?*

3.2.3 Multi-Modal Representation of Frames

After learning the objects and salient words, we represent each frame via the occurrence of salient words and objects. Formally, representation of the t^{th} frame of the i^{th} video is denoted as $\mathbf{y}_t^{(i)}$ and computed as $\mathbf{y}_t^{(i)} = [\mathbf{y}_t^{(i),1}, \mathbf{y}_t^{(i),v}]$ such that k^{th} entry of the $\mathbf{y}_t^{(i),1}$ is 1 if the subtitle of the frame has the k^{th} word and 0 otherwise. $\mathbf{y}_t^{(i),v}$ 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 $\mathbf{y}_t^{(i)}$. 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. $z_t^{(i)} \in \mathcal{N}$.

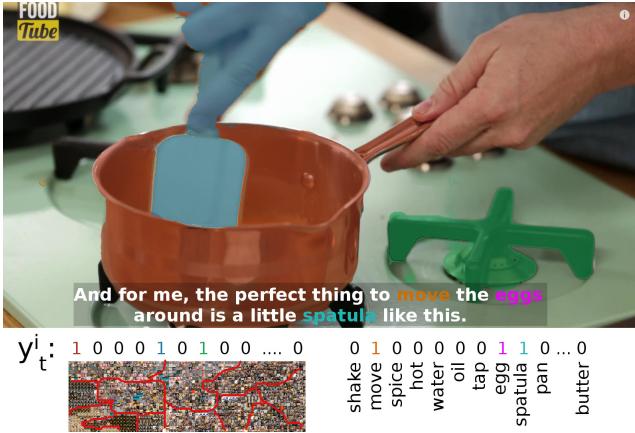


Figure 5: **Visualization of the representation of a sample frame.** 3 of the region proposals of the frame is included in the object clusters and 3 of the words in the subtitle of the frame is included in the salient word list.

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 $y_t^{(i)}$ is sampled from its activity distribution as $y_t^{(i)}|z_t^{(i)} = k \sim Ber(\theta_k)$. As a prior over θ , we use its conjugate distribution – *Beta distribution* –.

In the following sections, we first explain 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-Hastings samplers.

3.3.1 Beta Process Hidden Markov Model

For joint understanding of the time-series information, Fox et al.[8] 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;

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

where B_0 and the b_k are determined by the underlying Poisson process [10] 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 [10]. 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 $y_t^{(i)}$. Since we model each activity as a Bernoulli distribution, the emission probabilities follow the Bernoulli distribution as $p(y_t^{(i)}|z_t^{(i)}) = Ber(\theta_{z_t^{(i)}})$. Following the construction of the Fox et al.[8], 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

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

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

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.[8]. Our sampling procedure composed of iterative sampling of activity assignments ($\mathbf{f}^{(i)}$) from the current activity means θ_k , state assignments $z_t^{(i)}$ and observations $y_k^{(i)}$, and HMM parameters η, π, θ_k from the selected activities $\mathbf{f}^{(i)}$.

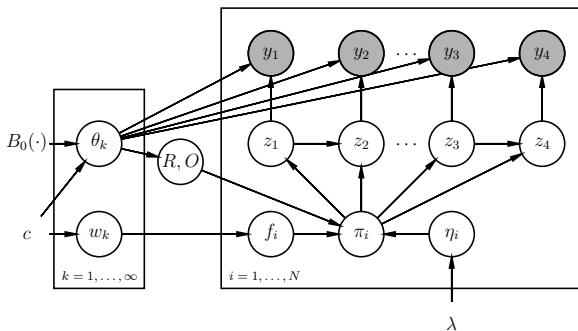
648
649
650
651
652
653
654
655
656
657
658
659
660

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 the each video becomes an Hidden Markov Model. See the text for the details.

We give the details of the sampler in the supplementary material.

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

4.1. Dataset

We guide our data collection effort with human preferences based on wikiHow [1] statistics. After obtaining the top100 queries people interested in wikiHow, we chose top25 ones which are directly related to the physical world and objects. We ignore the queries like *How to get over a break up?* and *How to write a resignation Letter?*. Resulting 25 queries are;

How to Bake Boneless Skinless Chicken, Cook Steak in a Frying Pan,

Make Jello Shots, Tell if Gold Is Real, Bake Chicken Breast, Hard Boil an Egg, Make Pancakes, Tie a Bow Tie Broil Steak, Make a Grilled Cheese Sandwich, Make Scrambled Eggs, Tie a Tie, Clean a Coffee Maker, Make a Milkshake, Make Yogurt, Unclog a Bathtub Drain, Cook an Omelet, Make a Smoothie, Poach an Egg, Cook Lobster Tails, Make Beef Jerky, Remove Gum from Clothes, Cook Ribs in the Oven, Make Ice Cream, Tell if an Egg is Bad

For each of the recipe, we queried YouTube and crawled the top 100 videos. We also downloaded the English subtitles if they exist. For evaluation set, we 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 fine-grained activities (*i.e.* steps of the recipe) as well as their labels. We also release the collected dataset at <http://anonymous.edu/MMRecipe>.

4.2. Implementation Details

Parameters:

Aligning Clusters: While comparing the results of our algorithm with the ground truth, we have an alignment problem. Our algorithm generates arbitrary IDs for clusters and the cluster IDs are not necessarily matching the ground truth IDs since the method is unsupervised. For example, we can name the activity 1 of ground truth as activity 3 although their content is same. So, we apply an alignment procedure and choose the alignment of cluster IDs which maximizes the intersection over union with the ground truth. We apply this method to all competing algorithms for fairness.

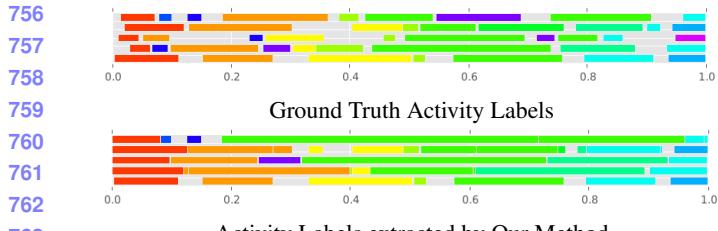
4.3. Qualitative Results

In this section, we visualize some of the results of our recipe understanding method. After running our algorithm 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 7a and 7b. 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 7a&7b, resulting clusters are semantically meaningful and correspond to the real activi-

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

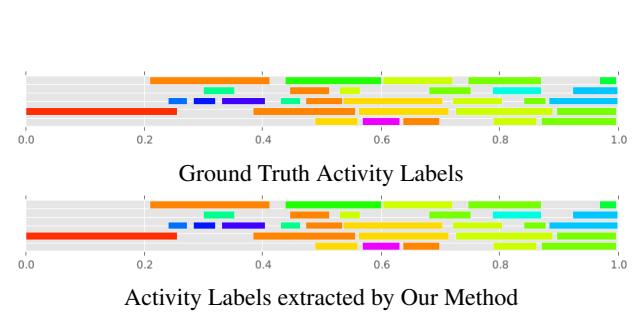


764 Crack the eggs one at a time
765 into a bowl.
766 Remove the omelet onto a
767 plate.
768



769 Eggs cook quickly, so make
770 You can either use a fork or sure the pan gets very hot
771 wire whisk to beat the eggs first; the butter melt com-
772 into a bowl.
773
774
775
776
777
778
779
780
781

(a) How to make an omelet?



819 Fill the blender to the first If you want a thicker milk-
820 shake, add more ice cream.
821



822 Mix the milkshake, first at Pour the milkshake into a
823 high speed then low. glass.
824

(b) How to make a milkshake?

825 over-segment the video and represent each segment by us-
826 ing state of the art low-level features from the activity de-
827 tection literature [?]. We are using dense trajectory features
828 for this purpose.

Category specific summary[29]: An algorithm proposed
829 by Potapov et al.[29] can detect the temporal boundaries
830 of the events/activities in the video from a time series data
831 without any supervision. It enforces a local similarity of
832 each resultant segment.

4.4.2 Metrics

Maximum Intersection over Union (IOU_{max}): In order
833 to evaluate the accuracy of the temporal segmen-
834 tation of the activities, we use intersection-over-union(IOU).
835 For N ground truth temporal activity segments (τ_i^* , $i \in N$), N' computed segments (τ_i' , $i \in N'$) and match-
836 ing function $m(\cdot)$ such that i^{th} ground truth segment is
837 matched to $m(i)^{th}$ computed segment, we define IOU as
838
$$IOU = \frac{1}{N} \sum_{i=1}^N \frac{\tau_i^* \cap \tau_{m(i)}'}{\tau_i^* \cup \tau_{m(i)}'}$$
. Since the matching function is
839 unknown in the supervised setting, we use the maximum
840 intersection-over-union while doing exhaustive search over
841

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

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

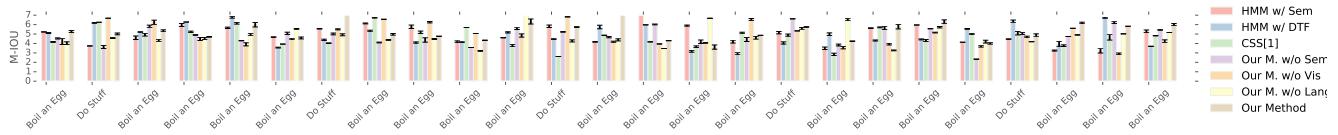
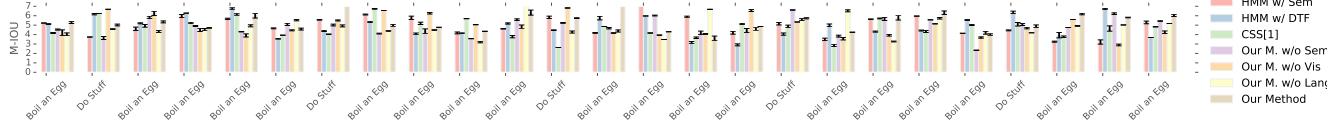
967

968

969

970

971

Figure 8: IOU_{max} stuff, not real numbers just a place holderFigure 9: AP_{max} stuff, not real numbers just a place holder

all matchings as; $IOU_{max} = \max_{m(\cdot)} \frac{1}{N} \sum_{i=1}^N \frac{\tau_i^* \cap \tau'_{m(i)}}{\tau_i^* \cup \tau'_{m(i)}}$

Maximum Average Precision (AP_{max}): Since the IOU_{max} is computed per video, it does not capture the accuracy of the detected activities over multiple videos. Hence, we also evaluate maximum average precision. 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 the same activities from multiple videos clustered into a single activity. Similarly, the maximum is over matching functions as $AP_{max} = \max_{m(\cdot)} \frac{1}{N} \sum_{i=1}^N \frac{tp_i}{tp_i + fp_i}$

4.4.3 Results

We discuss the quantitative and qualitative results in the light of the following questions.

Are the activities detected accurately? In this section, we discuss the results presented in Figure 8. Maximum-IOU captures the accuracy of the temporal segmentation of the videos. Since the ground-truth segmentation is the semantic one, high IOU_{max} requires both finding temporal activity boundaries and extracting correct activity definition. As shown in the Figure 8, proposed method consistently outperforms the competing algorithms. One interesting observation is the dramatic difference between the accuracy of HMM and our method. We believe this is the result of joint processing of multiple videos. HMM assumes all videos are generated from same set of activities with fixed transition probabilities and can not captures the inter-class variance. On the other hand, our algorithm is robust to inter-class variations since we are also modelling inclusion of activities for each video. Moreover, the segmentation problem is ill-posed since the granularity of the activities are subjective. Imposing the activity inclusion model brings an additional constraint to the problem and makes the problem well-posed as finding the set of small number of activities

which can construct the any video within the context of the recipe. In other words, the segmentation problem becomes learning the small dictionary of activities which is complete for the space of the recipe.

Are the same activities in different videos linked to each other? Although IOU_{max} successfully measures the accuracy of the detected activities, it can not measure the matching activities over different videos. Therefore, we are using AP_{max} for measuring the accuracy of matching different activities over multiple videos. AP_{max} is defined for each activity class and requires the algorithms to produce activity labels consistent with the ground truth.

How important is each modality? In order to experiment the importance of using both language and vision modalities, we compare our method with a self-baseline of using a single modality. As shown in Figure 8 and 9, our method significantly outperforms both of the baselines consistently in all recipes. Hence, we need to use both modalities. This result is expected because visual cues are good at separating different activities within the same video since the visual appearance is not changing much. However, language does not help much since there is too much background information other than the actual activity. On the other hand, language is good at relating activities from different videos since there is not much inter-class variation and it is easy to detect these variations caused by synonyms etc. thanks to the strong structure of the language modality.

References

- [1] wikiHow how to do anything. <http://www.wikihow.com>.
- [2] 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.

- 972 [3] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner,
973 D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates
974 making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages
975 529–536. IEEE, 2011. 2
- 976 [4] M. Bollini, J. Barry, and D. Rus. Bakebot: Baking cookies
977 with the pr2. In *The PR2 Workshop: Results, Challenges
978 and Lessons Learned in Advancing Robots with a Common
979 Platform, IROS*, 2011. 2
- 980 [5] J. Carreira and C. Sminchisescu. Constrained parametric
981 min-cuts for automatic object segmentation. In *Computer
982 Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010. 4
- 983 [6] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young,
984 C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture
985 tells a story: Generating sentences from images. In *Computer Vision–ECCV 2010*, pages 15–29. Springer, 2010.
986 2
- 987 [7] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth
988 a thousand pixels. In *Computer Vision and Pattern Recognition
989 (CVPR), 2013 IEEE Conference on*, pages 1995–2002. IEEE,
990 2013. 2
- 991 [8] E. Fox, M. Hughes, E. Sudderth, and M. Jordan. Joint modeling
992 of multiple related time series via the beta process with
993 application to motion capture segmentation. *Annals of Applied Statistics*, 8(3):1281–1313, 2014. 6
- 994 [9] F. Grabler, M. Agrawala, W. Li, M. Dontcheva, and
995 T. Igarashi. Generating photo manipulation tutorials by
996 demonstration. *ACM Transactions on Graphics (TOG)*,
997 28(3):66, 2009. 2
- 998 [10] T. Griffiths and Z. Ghahramani. Infinite latent feature models
999 and the indian buffet process. 2005. 6
- 1000 [11] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool.
1001 Creating summaries from user videos. In *Computer Vision–
1002 ECCV 2014*, pages 505–520. Springer, 2014. 2
- 1003 [12] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-
1004 S. Chua. Beyond search: event-driven summarization for
1005 web videos. *ACM Transactions on Multimedia Computing,
1006 Communications, and Applications (TOMM)*, 7(4):35, 2011.
1007 2
- 1008 [13] A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments
1009 for Generating Image Descriptions. *ArXiv e-prints*, Dec. 2014. 2
- 1010 [14] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-
1011 scale video summarization using web-image priors. In *Computer
1012 Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2698–2705. IEEE, 2013. 2
- 1013 [15] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of
1014 large-scale collections of web images and videos for story-
1015 line reconstruction. In *Computer Vision and Pattern Recog-
1016 nition (CVPR), 2014 IEEE Conference on*, pages 4225–4232.
1017 IEEE, 2014. 2
- 1018 [16] G. Kim and E. P. Xing. Reconstructing storyline graphs
1019 for image recommendation from web community photos.
1020 In *Computer Vision and Pattern Recognition (CVPR), 2014
1021 IEEE Conference on*, pages 3882–3889. IEEE, 2014. 2
- 1022 [17] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neu-
1023 ral language models. In *Proceedings of the 31st Interna-
1024 tional Conference on Machine Learning (ICML-14)*, pages
1025 595–603, 2014. 2
- 1026 [18] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What
1027 are you talking about? text-to-image coreference. In *Com-
1028 puter Vision and Pattern Recognition (CVPR), 2014 IEEE
1029 Conference on*, pages 3558–3565. IEEE, 2014. 2
- 1030 [19] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important
1031 people and objects for egocentric video summarization. In
1032 *CVPR*, volume 2, page 6, 2012. 2
- 1033 [20] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video
1034 object segmentation. In *Computer Vision (ICCV), 2011 IEEE
1035 International Conference on*, pages 1995–2002. IEEE, 2011.
1036 4
- 1037 [21] Z. Lu and K. Grauman. Story-driven summarization for
1038 egocentric video. In *Computer Vision and Pattern Recog-
1039 nition (CVPR), 2013 IEEE Conference on*, pages 2714–2721.
1040 IEEE, 2013. 2
- 1041 [22] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabino-
1042 novich, and K. Murphy. What’s Cookin’? Interpreting Cook-
1043 ing Videos using Text, Speech and Vision. *ArXiv e-prints*,
1044 Mar. 2015. 2
- 1045 [23] J. Malmaud, E. J. Wagner, N. Chang, and K. Murphy. Cook-
1046 ing with semantics. *ACL 2014*, page 33, 2014. 2
- 1047 [24] S. Mori, H. Maeta, Y. Yamakata, and T. Sasada. Flow graph
1048 corpus from recipe texts. In *Proceedings of the Ninth In-
1049 ternational Conference on Language Resources and Evalu-
1050 ation*, pages 2370–2377, 2014. 2
- 1051 [25] T. S. Motwani and R. J. Mooney. Improving video activ-
1052 ity recognition using object recognition and text mining. In
1053 *ECAI*, pages 600–605, 2012. 2
- 1054 [26] E. Olson, M. Walter, S. J. Teller, and J. J. Leonard. Single-
1055 cluster spectral graph partitioning for robotics applications.
1056 In *Robotics: Science and Systems*, pages 265–272, 2005. 4, 5
- 1057 [27] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: De-
1058 scribing images using 1 million captioned photographs. In
1059 *Advances in Neural Information Processing Systems*, pages
1060 1143–1151, 2011. 2
- 1061 [28] P. Perona and W. Freeman. A factorization approach to
1062 grouping. In *Computer VisionECCV’98*, pages 655–670.
1063 Springer, 1998. 5
- 1064 [29] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid.
1065 Category-specific video summarization. In *Computer
1066 Vision–ECCV 2014*, pages 540–555. Springer, 2014. 2, 8
- 1067 [30] L. R. Rabiner. A tutorial on hidden markov models and
1068 selected applications in speech recognition. In *PROCEED-
1069 INGS OF THE IEEE*, pages 257–286, 1989. 8
- 1070 [31] Y. Rui, A. Gupta, and A. Acero. Automatically extracting
1071 highlights for tv baseball programs. In *Proceedings of the
1072 eighth ACM international conference on Multimedia*, pages
1073 105–115. ACM, 2000. 2
- 1074 [32] R. Socher and L. Fei-Fei. Connecting modalities: Semi-
1075 supervised segmentation and annotation of images using un-
1076 aligned text corpora. In *Computer Vision and Pattern Recog-
1077 nition (CVPR), 2010 IEEE Conference on*, pages 966–973.
1078 IEEE, 2010. 2

- 1080 [33] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. 1134
1081 Ng. Grounded compositional semantics for finding and de- 1135
1082 scribing images with sentences. *Transactions of the Associa- 1136
1083 tion for Computational Linguistics*, 2:207–218, 2014. 2 1137
- 1084 [34] M. Tenorth, D. Nyga, and M. Beetz. Understanding and ex- 1138
1085 ecuting instructions for everyday manipulation tasks from the 1139
1086 world wide web. In *Robotics and Automation (ICRA), 2010* 1140
1087 *IEEE International Conference on*, pages 1486–1491. IEEE, 1141
1088 2010. 2 1142
- 1089 [35] B. T. Truong and S. Venkatesh. Video abstraction: A sys- 1143
1090 tematic review and classification. *ACM Transactions on 1144
1091 Multimedia Computing, Communications, and Applications*, 1145
1092 3(1):3, 2007. 2 1146
- 1093 [36] H. Yu and J. M. Siskind. Grounded language learning from 1147
1094 video described with sentences. In *ACL (1)*, pages 53–63, 1148
1095 2013. 2 1149
- 1096 [37] C. L. Zitnick and D. Parikh. Bringing semantics into fo- 1150
1097 cus using visual abstraction. In *Computer Vision and Pat- 1151
1098 tern Recognition (CVPR), 2013 IEEE Conference on*, pages 1152
1099 3009–3016. IEEE, 2013. 2 1153
- 1100 [38] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning 1154
1101 the visual interpretation of sentences. In *Computer Vi- 1155
1102 sion (ICCV), 2013 IEEE International Conference on*, pages 1156
1103 1681–1688. IEEE, 2013. 2 1157
- 1104 1158
- 1105 1159
- 1106 1160
- 1107 1161
- 1108 1162
- 1109 1163
- 1110 1164
- 1111 1165
- 1112 1166
- 1113 1167
- 1114 1168
- 1115 1169
- 1116 1170
- 1117 1171
- 1118 1172
- 1119 1173
- 1120 1174
- 1121 1175
- 1122 1176
- 1123 1177
- 1124 1178
- 1125 1179
- 1126 1180
- 1127 1181
- 1128 1182
- 1129 1183
- 1130 1184
- 1131 1185
- 1132 1186
- 1133 1187