

Automatic annotation of unique locations from video and text

Chris Engels¹

Chris.Engels@esat.kuleuven.be

Koen Deschacht²

Koen.Deschacht@cs.kuleuven.be

Jan Hendrik Becker¹

JanHendrik.Becker@esat.kuleuven.be

Tinne Tuytelaars¹

Tinne.Tuytelaars@esat.kuleuven.be

Marie-Francine Moens²

Sien.Moens@cs.kuleuven.be

Luc Van Gool¹³

Luc.VanGool@esat.kuleuven.be

¹ ESAT-PSI

K.U. Leuven

² Department of Computer Science

K.U. Leuven

³ Computer Vision Laboratory

BIWI/ETH Zürich

Abstract

Given a video and associated text, we propose an automatic annotation scheme in which we employ a latent topic model to generate topic distributions from weighted text and then modify these distributions based on visual similarity. We apply this scheme to location annotation of a television series for which transcripts are available. The topic distributions allow us to avoid explicit classification, which is useful in cases where the exact number of locations is unknown. Moreover, many locations are unique to a single episode, making it impossible to obtain representative training data for a supervised approach. Our method first segments the episode into scenes by fusing cues from both images and text. We then assign location-oriented weights to the text and generate topic distributions for each scene using Latent Dirichlet Allocation. Finally, we update the topic distributions using the distributions of visually similar scenes. We formulate our visual similarity between scenes as an Earth Mover's Distance problem. We quantitatively validate our multi-modal approach to segmentation and qualitatively evaluate the resulting location annotations. Our results demonstrate that we are able to generate accurate annotations, even for locations only seen in a single episode.

1 Introduction

In this paper, we tackle the challenging problem of extracting information from unstructured text and exploiting this information to annotate an associated video. In particular, we develop a method that operates on a video with associated transcript, segments the video into scenes that are set in a specific location, and automatically annotates each scene with a textual label that provides a description of that location (*e.g.* “Joyce’s living room”). These labels are

not derived from external training information, and we do not need to know the number of locations in a given video. The resulting location descriptions may be used in an information retrieval system, presented to end-users or used to train an automatic location classifier.

As a test set, we focus on action series, which present many challenges for automated location annotation. Most substantially, many locations are unique to specific episodes. Thus, any supervised approach that assumes annotations from other episodes is guaranteed to fail. Other episodes can only be used to learn global statistics and parameter settings. For the actual location labeling, only the transcript of the episode being annotated is utilized. Other problems for visual recognition in this context include rapid camera switching and motion, nondescript or blurred backgrounds, and individuals appearing in multiple locations.

We tackle this challenging problem in the following way: We begin by roughly aligning the transcript to the video using subtitles. We then refine this alignment and split the text and video of the episode into scenes using both modalities (Sec. 3). The core of this article is the automatic generation of location labels for all scenes (Sec. 4). The transcripts describe many of the locations within each episode, but this information may be hidden within the text or not present for many scenes. The solution we propose is based on generating textual labels using a weighted topic mixture model. We first rely on an automatic detector to find location descriptions in the text. These detected locations are used to learn topic mixtures for each scene with Latent Dirichlet Allocation (LDA). Using this topic model allows us to use contextual information when labeling a location (*e.g.* “sink” as an implicit cue for the location “kitchen”) and to handle different wordings for the same location. We then modify the topic distribution for each scene using a visual similarity measure based on Earth Mover’s Distance; this step propagates labels to scenes lacking informative text. Finally, we present quantitative results and a subjective human evaluation in Sec. 5.

2 Previous Work

Generic scene type classification, which seeks to describe the kind of location seen in an image (*e.g.* “beach” or “street”) has been studied *e.g.* in [12, 21]. Such approaches mostly rely on supervised techniques and large sets of annotated training data.

Several papers have proposed weakly supervised methods focusing on automatic video annotation. Schaffalitzky and Zisserman [19] retrieve images of a particular location based on wide baseline matching techniques. Héritier *et al.* [10] use latent topic models to identify discriminative and often reoccurring parts of locations using SIFT features, which are then labeled manually. Neither of these works aims at a full annotation of each scene occurring in a video, nor do they explore the use of complementary text to obtain a fully automatic pipeline. Zhu and Liu [24] study the problem of segmentation into scenes and classify the obtained scenes into either conversation, suspense, or action scenes, based on audio and video and using heuristic rules for the actual classification. Finally, Zhai and Shah [23] perform scene segmentation based on a purely visual Markov chain Monte Carlo approach, without attempting to classify the obtained scenes in any way.

Other authors have looked into the use of readily available textual annotation for TV and movie footage to learn to annotate in a weakly supervised manner. In particular, Cour *et al.* [8] propose a unified generative model that integrates scene segmentation, script alignment, and shot threading. Everingham *et al.* [9] use transcripts aligned to the video data based on the subtitles to then identify the cast in a soap series. Laptev *et al.* [11] exploit scripts for action recognition in Hollywood movies, using a supervised text classifier and

using a kernel-based discriminative clustering algorithm to overcome problems with inaccurate alignment between video and text [6]. Finally, Marszałek *et al.* [24] rank video segments based on actions using mining techniques. They also extract location names from the scripts, but they rely on specifically-formatted scripts and do not focus on specific locations, but on scene types (*e.g.* “exterior location”).

Several works, *e.g.* [2, 13, 16], have investigated the use of cross-modal topic models in the context of automatic image annotation to fuse visual and textual information. However, it turns out it is relatively difficult to balance the contributions of both modalities. Moreover, in our application, text and visual information are only weakly linked, often with complementary information present in only one of the two modalities.

Instead, we use Latent Dirichlet Allocation (LDA) [3] to generate labels from the text alone. LDA generatively models each document as a multinomial mixture of latent topics from which the actual words are drawn. The parameters of the multinomial topic distributions of each document are sampled from a Dirichlet prior. The generative process can be summarized as follows: for each document d a multinomial mixture parameter θ is sampled first. Second, for each word w a topic z is sampled from the multinomial distribution and, third, the word w is sampled from the multinomial word distribution conditioned on that topic. Using variational Bayes inference, the topic distributions within each document are inferred from the observed words.

We finally incorporate the visual information by updating the textual topic distributions based on visual similarity. This is, in some sense, similar to the tag propagation proposed by Guillaumin *et al.* [9].

3 Scene Segmentation and Alignment

We first segment the video and transcripts into scenes. We assume each scene occurs within a single location and can therefore be described by a single location label.

We adopt a multi-modal approach to locate the scene cuts by integrating cues from both video and text. Because the transcript does not contain any timing information, we first need to temporally align it to the video. We then learn the probability that a sentence describes a change in the location. Next, we detect visual shot cuts, which precisely localize possible scene boundaries. Finally, we iteratively merge shots using the text for guidance.

We shall employ the following notation: we define a sentence $\mathbf{w} = \{w_1, \dots, w_n\}$, where w is a word; $\hat{\mathbf{w}} = \langle \mathbf{w}, t_{\mathbf{w}} \rangle$ denotes a sentence at time $t_{\mathbf{w}}$; and a scene $s = \langle I, W \rangle$, where I is a time interval and $W = \{\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \dots, \hat{\mathbf{w}}_m\}$ is the set of sentences occurring within I .

3.1 Coarse Alignment

As seen in Fig. 1, the transcripts provide a natural language description of each episode including the spoken dialog. However, they contain no temporal information for the descriptions or dialog. We use the time-warping approach of Everingham *et al.* [7] to obtain an initial alignment between the dialog in the transcripts and the subtitles from the video.

While the dialog alignment is reasonably precise, the alignment of the textual descriptions is significantly harder. The dialog timing bounds the possible timing of descriptive sentences, but there may be several such sentences between spoken lines. In action scenes with limited speech, the alignment of the descriptions becomes merely approximate.

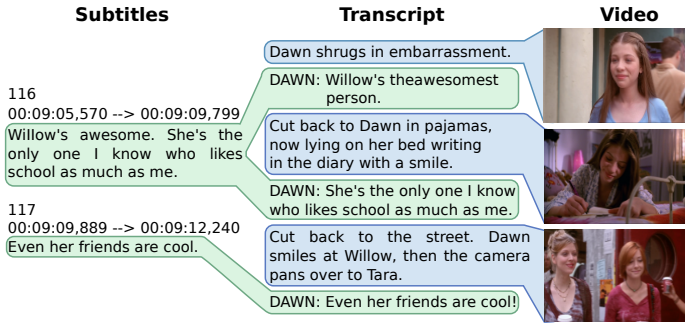


Figure 1: Example of transcript to video alignment.

We estimate the sentence time t_w by interpolating between the nearest surrounding subtitle times. If there are multiple sentences between subtitles, they are distributed evenly within the time period. Instead of associating a cut directly with that time, we learn a mean offset μ_{text} and standard deviation σ_{text} for the times from training episodes.

Detecting visual shot cuts Shot boundary detection in video is fairly well-established; see *e.g.* Yuan *et al.* [22] for a comprehensive review. Our implementation uses a sliding window over color histograms to compute a dissimilarity energy based on χ^2 -distance, followed by local nonmaximum suppression and thresholding. We use the detected shot cuts $T_{\text{cut}} = \{t_{\text{cut}}\}$ below for scene construction.

3.2 Learning scene cuts in the text

The descriptive part of the transcripts often contains strong cues for the start of new scenes, *e.g.* “*Fade in on a beach, daytime.*”. The dialog usually contains no such clues, so we discard it for the purpose of learning cuts. We preprocess the text by dividing the textual descriptions into sentences, tokenizing sentences into words, performing part-of-speech tagging (using the LTPOS tagger [15]), and reducing word forms to their lemmas (*e.g.* ‘running’ becomes ‘run’). Finally, we train a supervised Naive Bayes classifier on a training set of three manually annotated episodes. For each sentence \mathbf{w} , the classifier computes the probability $p_{\text{cut}}(\mathbf{w})$ that the sentence describes a scene cut.

For $\hat{\mathbf{w}} = \langle \mathbf{w}, t \rangle$, we approximate the probability that \mathbf{w} describes a cut at time t as

$$p_{\text{cut}}(\hat{\mathbf{w}}) = p_{\text{cut}}(\mathbf{w}) \mathcal{N}(t | (t_w + \mu_{\text{text}}), \sigma_{\text{text}}), \quad (1)$$

where $\mathcal{N}(x | \mu, \sigma)$ is a Gaussian distribution evaluated at x .

3.3 Agglomerative scene construction

Many scenes feature dialog consisting shots switching between two or more unique perspectives, so we need to be robust to such cases. Additionally, we want to integrate the cues from the transcripts learned by the classifier. Similarly to agglomerative clustering, we approach the construction of scenes in a bottom-up manner by iteratively merging shots that belong to the same scene.

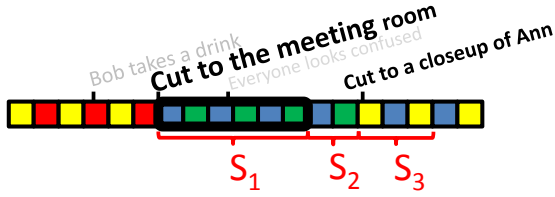


Figure 2: Toy example of scene merging. Each box corresponds to a shot, and matching colors imply similar appearance. Larger text corresponds to higher probability of a scene cut. s_1 has already been partially merged, while s_2 and s_3 are hypothesized scenes. $d_{\text{scene}}(s_1, s_2) < d_{\text{scene}}(s_1, s_3)$, so s_2 would be more likely to merge with s_1 than s_3 .

We first need a measure of visual dissimilarity for comparing two sets of shots. Several works have noted that Earth Mover’s Distance (EMD) provides an efficient metric for comparing images [18] or video clips. Our approach is similar to that of Peng and Ngo [17], who rank video clip similarity by measuring the EMD between the shots in two clips. They use color histograms to describe shots, construct a distance matrix from histogram intersection, and define weights from the number of frames. The resulting measure gives a many-to-many comparison that is robust to poor segmentation and small outlier observations.

In our case, we first define a distance between shots. We represent each video shot by its mean color histogram and number of frames. Our color histograms are generated from k-means clustering in CIELab color space. As suggested by Rubner *et al.* [18], we define the distance between colors \mathbf{c}_m and \mathbf{c}_n as:

$$d_{\text{color}}(\mathbf{c}_m, \mathbf{c}_n) = 1 - \exp(-\alpha \|\mathbf{c}_m - \mathbf{c}_n\|_2), \quad (2)$$

where $\alpha = \|\left[\sigma_L \sigma_a \sigma_b\right]^T\|_2$, and σ_* denotes the standard deviation of each color channel. We compute the EMD d_{shot} between two shots using the histograms to define weights and d_{color} as the ground distance. We then assume a scene is composed of some number of shots and adopt a similar strategy to Peng and Ngo [17] mentioned above. We use d_{shot} as a ground distance and the number of frames in each shot as weights. However, we modify the distance of adjacent scenes s_i, s_{i+1} using the probability of a text cut occurring within those scenes:

$$d_{\text{scene}}(s_i, s_{i+1}) = \frac{\text{EMD}(s_i, s_{i+1})}{\prod_{t_{\text{cut}} \in (I_i \cup I_{i+1}) \cap T_{\text{cut}}} 1 - \max_{\mathbf{w}_i} p_{\text{cut}}(\langle \mathbf{w}_i, t_{\text{cut}} \rangle)}. \quad (3)$$

Note that the term within the denominator is only evaluated at shot cut detections lying strictly within the interval. This inhibits merging over previously detected cuts.

We initialize the merging process by defining each video shot as its own scene. At each iteration, we merge neighboring scenes where d_{scene} is minimal. Instead of considering only immediate pairs of neighbors, we also hypothesize and evaluate larger scenes within a neighborhood. In other words, we compute not only $d_{\text{scene}}(s_i, s_{i+1})$, but also *e.g.* $d_{\text{scene}}(s_i \cup s_{i+1} \cup s_{i+2}, s_{i+3} \cup s_{i+4})$, up to some maximum neighborhood (in our experiments, we combine up to three scenes in a hypothesis). These expanded scenes help us both to capture dialogs or similar patterns of shots and to quickly merge very similar adjacent shots. Fig. 2 shows a few relevant examples. s_1 is a scene merged in a previous iteration, while s_2 and s_3 are hypothetical scenes. Individually, the shots in s_2 appear dissimilar. Taken in context of

other shots in s_1 , they more clearly belong to the same scene, resulting in a lower value for $d_{\text{scene}}(s_1, s_2)$ than the distance between the two shots in s_2 .

In order to prevent under-segmentation, we need to stop merging at an appropriate number of scenes. To do so, we learn a threshold on d_{scene} that enforces a minimum recall on detected ground truth cuts in training episodes.

4 Location annotation

The textual descriptions of locations contain significant variation, posing several problems for learning location labels for scenes. Problems often arise from synonymy and polysemy, where multiple phrases are used to refer to the same location (e.g. “cemetery” and “graveyard”), or different locations are referred to by the same phrase (e.g. “the room”), respectively. Alternatively, a location may not be mentioned explicitly within a scene, although in some cases it could be inferred from other contextual cues (e.g. “fridge” in the kitchen).

A topic model could help address these problems by distributing words related to different locations into separate topics. However, there is no intrinsic reason that location descriptions should appear prominently within the topic distributions, since the standard LDA framework only uses term counts as weights. In order to increase this prominence, we modify the term weighting of the input text such that we reduce the influence of terms that are less indicative of location and bias the topics toward locations.

Furthermore, some scenes lack any descriptive text, so their respective topic distributions will not be useful in selecting a label. To overcome this issue, we use visually similar scenes to propagate critical words to the ambiguous scenes.

4.1 Location phrases based on text

Location phrases $\tilde{\mathbf{w}}_i = \{w_i, \dots, w_{i+n}\}$, with $n \geq 0$, are a word or a sequence of words that describe a location.

To identify phrases that express a location, each word w_i is labeled as "I" (Inside) or "O" (Outside a location phrase). We then consider each sequence of words that all are labeled with "I" and are delineated by "O" labels as a location phrase. An alternative is to consider also a third label "B" (Begin) in addition to the "I" and "O" labels. "B" then indicates the word with which the location phrase starts. This popular "BIO" encoding resulted in a lower performance than the simple "IO" labeling and was abandoned in our experiments. This method for selecting location phrases does not place a limit on the length of the location phrase, and does not rely on a syntactic sentence parser to segment a sentence into several phrases beforehand. To automatically detect the location phrases, we use the standard hidden Markov model [10]. This model is trained on manually annotated texts and assigns a probability $p_{\text{loc}}(\tilde{\mathbf{w}}_i)$ to every phrase $\tilde{\mathbf{w}}_i = \{w_i, \dots, w_{i+n}\}$ in an unseen text. Every word is represented by the following features: the word token, a list of synonyms for the word that is automatically learned [11] from the Reuters corpus, and $p_{\text{cut}}(\mathbf{w})$. The last feature is motivated by the fact that the location is often described in the sentence that contains the scene cut, e.g. “Cut to the kitchen”.

To estimate the LDA topic mixtures, all phrases $\tilde{\mathbf{w}}_i$ in a scene are weighted by their probability $p_{\text{loc}}(\tilde{\mathbf{w}}_i)$. We found that assigning at least a small positive value to every word helps to disambiguate locations by providing more context. For each scene, we collect the weighted counts of all phrases, which is then considered a single document for which LDA learns a

topic mixture. We empirically found that the quality of topic distributions is relatively stable over a reasonable range of number of topics selected between the actual numbers of locations and scenes. In our experiments, we choose $k = 35$ topics.

4.2 Visual similarity

Given two scenes, we need to compute a measure for the visual similarity of the scene locations. In the foreground of a typical scene, there are often one or more persons present. The background may be cluttered, out of focus, sparsely detailed, and occluded by people. Additionally, the camera perspective may be stationary, move smoothly, or frequently cut away. Persons themselves are not indicative of a certain location, as they may appear in different locations. Therefore, we use the upper-body pose detector of Ferrari *et al.* [8] to excise them from the scenes as much as possible, prior to computing a visual scene descriptor for the scene similarity measure.

We compute scene distances similarly to those of the segmentation step. However, we modify the color histograms to not include pixels from detected poses. For efficiency, we group the shots using spectral clustering and use the mean histogram and total number of frames in the cluster to compute the distance.

We convert the distance between scenes s and s' into a similarity matrix:

$$A(s, s') = \exp(-\lambda \text{EMD}(s, s')^2), \quad (4)$$

where λ is a scaling parameter which we determine from training data.

4.3 Combination of vision and text

Regularly, scenes do not contain a location phrase in the transcript, resulting in topic distributions for these scenes that contain little information on the location. To address this problem we combine the topic distribution of a scene with the topic distributions of visually similar scenes:

$$\tilde{p}(z_i|s) = \frac{1}{2}(p(z_i|s) + \sum_{s' \in S \setminus s} \pi(s, s')p(z_i|s')). \quad (5)$$

The mixing coefficients $\pi(s, s')$ are given by the normalized visual similarity between scenes s and s' :

$$\pi(s, s') = \frac{A(s, s')}{\sum_{s' \in S \setminus s} A(s, s')}. \quad (6)$$

This method effectively allows to propagate location labels between visually similar scenes, especially when scenes do not contain a location phrase in the transcript.

The final location label $\tilde{\mathbf{w}}_s^*$ for a scene s is selected as the location phrase $\tilde{\mathbf{w}}_i$ with maximum probability, given the number of topics k and reweighted topic distribution $\tilde{p}(z_i|s)$:

$$\tilde{\mathbf{w}}_s^* = \arg \max_{\tilde{\mathbf{w}}_i} \prod_{j=1}^k p(\tilde{\mathbf{w}}_i|z_j)\tilde{p}(z_j|s) \quad (7)$$

5 Evaluation

Unlike supervised methods that output a unique class for a given input, our system generates a list of phrases and corresponding probabilities, with the most likely phrase being assigned as the label. Because these phrases are generated within the context of the transcript, multiple phrases could be considered valid. For example, “kitchen”, “Joyce’s kitchen” and “interior of the kitchen” may all be used to refer to the same location. This polysemy means that creating a useful ground truth for automatic evaluation of the final result is nontrivial, so we instead rely on a qualitative evaluation to validate our approach. Where training data is required, we test our system using leave-one-out cross validation at the episode level.

Because of the large amount of effort required for manual evaluation, we limit our test set to four episodes of *Buffy the Vampire Slayer* (Season 5, Episodes 1 – 4), for which fan-generated transcripts are available online [24]. These episodes provide a challenging validation for our system due to unstructured transcripts, highly variable lighting conditions, frequent motion blurring and shot cuts, and the diversity of locations in each episode. Each episode has on average 53 scenes, and from a total of 64 individual locations only 18 are shared across episodes. Given these statistics, it is clear that a supervised approach trained on other episodes alone is bound to fail.

5.1 Scene cuts

We first note the performance of our approach to scene cut generation, for which ground truth evaluation is possible. The text-based scene cut classifier achieves a leave-one-out cross validation precision of 91.57%, recall of 77.95% and F1-score of 84.21%. The lower recall is largely caused by an incorrect classification of sentences that describe the actors moving from one location to another, e.g. “*Buffy goes into another room.*”. We consider such cases to be scene cuts because the location changes, even though there may be a continuous transition.

After agglomerative clustering, the scene cuts achieve an average precision of 47.94% for a recall of 80.50% when allowing an offset of up to 50 frames (2 seconds). Effectively, the approach limits incorrectly merged scenes at the cost of segmenting scenes twice as often as necessary. Many of the missed scene cuts are in areas with either few text descriptions or dialog, respectively leading to low cut probabilities or imprecise cuts.

5.2 Location phrase detection in text

As described in Sec. 4.1, we train a hidden Markov model on three annotated episodes and apply it on a fourth episode, achieving on average a precision of 77.45%, a recall of 76.47% and an F1-score of 76.96%. The most common source of error by the model is incorrect segmentation, where only part of a location is correctly labeled (e.g. labeling “Giles’ ” instead of “Giles’ apartment”).

5.3 Location labels

To evaluate the final labels, we provided thirteen human evaluators with image frames and transcript text for each estimated scene in order to provide them with context of the location. They were presented with labels generated from three different methods and asked to decide whether the labels accurately describe the location of the scene. The methods select the most






				
0.011 ball	0.714 dawn's_room	0.002 looking	0.001 frowns	0.928 sesame street
0.003 the ball	0.014 diary	0.001 in the	0.001 eyes	0.272 dracula's accent
0.001 grinning	0.001 writing	0.001 confused	0.001 asleep	0.017 street
0.048 beach	0.348 dawn's room	0.195 the bedroom	0.158 exterior shot	0.431 sesame street
0.047 hall	0.068 a bedroom	0.124 joyce's room	0.142 outside	0.061 dracula's accent
0.040 the kitchen	0.057 bedroom	0.073 the kitchen	0.076 giles' building	0.035 exterior shot

Figure 3: Examples of correct annotations (columns 1-3) and incorrect annotations (4-5). Top: representative frames. Middle: most likely labels given only text. Bottom: labels after vision-based update, with final annotation in bold. Each label is shown with its probability.

likely location phrase either from the text within a single scene (*text*), the topic distribution generated in Sec. 4.1 (*text+lda*), or the visually-updated distributions described in Sec. 4.3 (*text+lda+vision*).

We score each episode by the percentage of time (as opposed to scenes) that the label is correct. We report the average accuracy of each system, along with the standard deviation, in Table 1. Both *text+lda* and *text+lda+vision* improve upon the *text* method, while the strongest performance increase comes from adding the vision component. Fig. 3 shows several examples of the most likely location labels before and after the vision-based update, as well as their respective probabilities. The first three columns show how the update reinforces the correct labels, especially when no clear label is extracted from the text. The last two columns show possible failure cases of our approach. In the fourth column, no location label was detected in the text, and the imagery was noninformative. In these cases, the updated distribution tends to the most likely phrases over the entire episode. In the last column, an incorrect label was detected in the text.

6 Conclusion

We proposed a novel multi-modal system for automatic annotation of locations from video and text and demonstrated its performance on an action series. Despite the challenges posed by the limited amount of unstructured text description in the transcripts, we successfully detect scene cuts and are able to annotate even locations that are unique to single episodes. The use of a topic model at an intermediate stage increases the accuracy of the location annotations. Our visual similarity measure allows us to evolve from shot cuts to scene cuts and improve the location annotations by propagating labels between visually similar scenes.

episode	text	text+lda	text+lda+vision
1	58.38% \pm 4.88%	61.49% \pm 4.88%	67.62% \pm 7.85%
2	64.37% \pm 8.23%	67.87% \pm 7.91%	75.72% \pm 6.76%
3	69.86% \pm 3.60%	69.55% \pm 5.96%	71.43% \pm 6.97%
4	53.36% \pm 4.11%	55.46% \pm 4.81%	63.02% \pm 10.26%
all	61.50% \pm 8.24%	63.59% \pm 8.11%	69.45% \pm 9.16%

Table 1: Accuracy and standard deviation of the proposed systems.

Acknowledgments

We wish to thank Marcin Eichner and Vittorio Ferrari for their assistance with creating pose estimations. This work was supported by the IWT project AMASS++ (SBO-060051), ERC grant COGNIMUND, and the European IST Programme CLASS Project (FP6-027978).

References

- [1] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.
- [2] David M. Blei and Michael I. Jordan. Modeling annotated data. In *International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 2003.
- [3] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [4] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *European Conference on Computer Vision*, pages 158–171, Berlin, Heidelberg, 2008. Springer-Verlag.
- [5] K. Deschacht and M. Moens. Semi-supervised semantic role labeling using the Latent Words Language Model. In *Conference on Empirical Methods in Natural Language Processing*, 2009.
- [6] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *IEEE International Conference on Computer Vision*, 2009.
- [7] M. Everingham, J. Sivic, and A. Zisserman. Hello! My name is... Buffy – automatic naming of characters in TV video. In *British Machine Vision Conference*, volume 2, 2006.
- [8] V. Ferrari, M. Marín-Jiménez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [9] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *IEEE International Conference on Computer Vision*, 2009.
- [10] M. Hérítier, S. Foucher, and L. Gagnon. Key-places detection and clustering in movies using latent aspects. In *International Conference on Image Processing*, 2007.
- [11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [12] L. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2006.

- [13] L. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [14] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [15] A. Mikheev. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423, 1997.
- [16] F. Monay and D. Gattica-Perez. On image auto-annotation with latent space models. In *ACM Multimedia*, pages 275–278, 2003.
- [17] Y. Peng and C. Ngo. Emd-based video clip retrieval by many-to-many matching. In *International Conference on Image and Video Retrieval*, pages 71–81, 2005.
- [18] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:2000, 2000.
- [19] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92(2-3):236–264, 2003.
- [20] Twiz TV. Buffy, The Vampire Slayer Transcripts. <http://www.twiztv.com/scripts/buffy/>.
- [21] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007.
- [22] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A formal study of shot boundary detection. In *IEEE Transactions on Circuit and Systems For Video Technology*, pages 168–186, 2007.
- [23] Y. Zhai and M. Shah. A general framework for temporal video scene segmentation. In *IEEE International Conference on Computer Vision*, volume 2, 2005.
- [24] S. Zhu and Y. Liu. Video scene segmentation and semantic representation using a novel scheme. *Multimedia Tools Appl.*, 42(2):183–205, 2009. ISSN 1380-7501.