Finding Actors and Actions in Movies

P. Bojanowski[2,*] F. Bach[2,*] I. Laptev[2,*] J. Ponce[1,*] C. Schmid[2,†] J. Sivic[2,*]
[1]École Normale Supérieure [2]INRIA

Abstract

We address the problem of learning a joint model of actors and actions in movies using weak supervision provided by scripts. Specifically, we extract actor/action pairs from the script and use them as constraints in a discriminative clustering framework. The corresponding optimization problem is formulated as a quadratic program under linear constraints. People in video are represented by automatically extracted and tracked faces together with corresponding motion features. First, we apply the proposed framework to the task of learning names of characters in the movie and demonstrate significant improvements over previous methods used for this task. Second, we explore the joint actor/action constraint and show its advantage for weakly supervised action learning. We validate our method in the challenging setting of localizing and recognizing characters and their actions in feature length movies Casablanca and American Beauty.

1. Introduction

The recognition of actions, scenes and objects in videos is a difficult task due to the large variability of their visual appearance. Modeling such a variability typically requires manually annotating large numbers of training samples for learning model parameters. Video annotation, however, is a very tedious process that does not scale well to the huge number of existing events.

Video scripts exist for thousands of movies and TV-series and contain rich descriptions in terms of people, their actions, interactions and emotions, object properties, scene layouts and more. Previous work has explored video scripts to learn and automatically annotate characters in TV series [6, 22, 23]. Automatic learning of human actions from scripts has also been attempted [8, 18, 20]. The problem, however, remains difficult due to the lack of explicit correspondence between scene elements in video and their textual descriptions in scripts. In particular, video scripts provide no spatial localization of people and objects, and the temporal localization of events inferred from the subtitles is often imprecise.

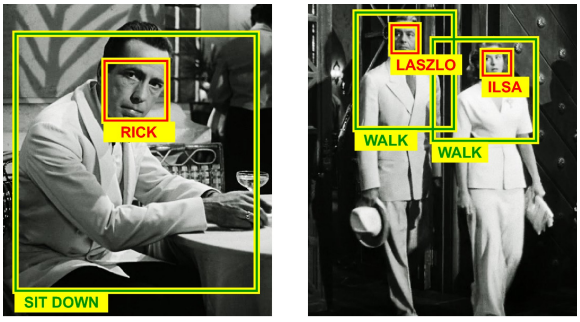


Figure 1: Result of our automatic detection and annotation of characters and their actions in the movie Casablanca. The automatically resolved correspondence between video and script is color-coded.

Previous work on weakly supervised learning in images [5, 13, 19] and video [6, 8, 18, 20, 22, 23] has explored redundancy to resolve ambiguity of textual annotation. For example, multiple images known to contain a person X could help identifying X by intersecting sets of people from each image. Given the difficulty of identifying the same person, action or object class in different images or videos, the realization of the "intersection" idea, however, is often non-trivial in practice.

Objects, people and actions often co-occur. Knowing that "Rick sits down" in a video can help annotating a sitting down action if we can localize Rick and vice versa, see Figure 1. Action recognition can particularly help person identification for rare subjects and subjects facing away from the camera (e.g., Ilsa walks away to the door). Recognizing actors, on the other hand, can be most useful for learning rare events (e.g. hand shaking).

We follow this intuition and address *joint* weakly supervised learning of actors and actions by exploiting their co-occurrence in movies. We follow previous work [6, 8, 18, 20, 22, 23] and use movie scripts as a source of weak supervision. Differently from this prior work, we use actor-action

*WILLOW project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France.
†LEAR team, INRIA Grenoble Rhône-Alpes, Paris, France

co-occurrences derived from scripts to constrain the weakly supervised learning problem.

As one of our main contributions, we formulate weakly supervised joint learning of actors and actions as an optimization of a new discriminative cost function. We first investigate weakly supervised learning of actors only and demonstrate the benefit of our learning method in comparison with other weakly supervised techniques designed for this task [6, 22]. We then demonstrate the advantage of the joint constraints for action recognition. We validate our method in the challenging setting of localizing and recognizing actors and their actions in movies Casablanca and American Beauty. An example output of our algorithm for a short movie clip and the associated script section is illustrated in Figure 1.

Related Work. Learning from images and text has been addressed in the context of automatic annotation of images with keywords [4, 11, 25] or labeling faces with names in news collections [5]. Berg *et al.* [5] label detected faces in news photographs with names of people obtained from text captions. Recent work has looked at learning spatial relations (such as "on top of") from prepositions [13] or generating entire sentence-level captions for images [10, 21]. A generative model of faces and poses (such as "Hit Backhand") was learnt from names and verbs in manually provided captions for news photographs [19]. While the goal of this work is related to ours, we focus on learning from video with sparse, noisy and imprecise annotations extracted from scripts. To deal with the ambiguity of annotations, we develop a new discriminative weakly supervised clustering model of video and text.

In video, manually provided text descriptions have been used to learn a causal model of human actions in the constrained domain of sports events [14]. Others have looked at learning from videos with readily-available text, but names [6, 9, 22] and actions [8, 18] have been so far considered separately. The ambiguity and errors of readily available annotations present a major challenge for any learning algorithm. These problems have been addressed by designing appropriate loss functions [6] or explicitly finding the corresponding instances in video using multiple instance learning [8]. Others have looked at convex relaxations of discriminative clustering with hidden variables for image co-segmentation [16, 17].

Contributions. First, we consider a richer use of textual information for video and learn from pairs of names and actions co-occurring in the text. Second, we formulate the problem of finding characters and their actions as weakly supervised structured classification of pairs of action and name labels. Third, we develop a new discriminative clustering model jointly learning both actions and names and incorporating text annotations as constraints. The corresponding optimization is formulated as a quadratic program under linear constraints. Finally, we demonstrate the validity of the model on two feature-length movies and corresponding movie scripts, and demonstrate improvements over earlier weakly supervised methods.

2. Joint Model of Actors and Actions

We formulate the problem of jointly detecting actors and actions as discriminative clustering [2, 17]: grouping samples into classes so that an appropriate loss is minimized. We incorporate text-based knowledge as a suitable set of constraints on the cluster membership.

2.1. Notation

Let us suppose that we have two label sets $\mathcal{P}$ and $\mathcal{A}$ and that $|\mathcal{P}| = P$ and $|\mathcal{A}| = A$. In practice, one can think of these as person and action classes.

Our data is organized into sets, that we refer to as bags, and which are indexed by $i \in I$. Every bag has a set of samples $\mathcal{N}_i$ and a set of annotations Λ_i. In our application, $\mathcal{N}_i$ is the group of person tracks appearing in a scene while Λ_i can be thought of as a set of sentences specifying who is doing what. In the following, we write $N = \sum_{i \in I} |\mathcal{N}_i|$.

For every sample $n \in \mathcal{N}_i$ we have a feature vector $x_n \in \mathbb{R}^{1 \times d}$, and some latent variables detailed next. Every sample belongs to a class in $\mathcal{P}$ and a class in $\mathcal{A}$. For each sample we therefore define a pair of latent variables z_n in $\{0,1\}^{1 \times P}$ and t_n in $\{0,1\}^{1 \times A}$ indicating to which person and action class it belongs. We define X to be a $N \times d$ data matrix with rows x_n, Z is a $N \times P$ matrix with person labels in rows z_n and T is a $N \times A$ matrix with action labels in rows t_n. The p-th element of a vector z_n is written z_{np}.

Given weak supervision in the form of constraints on Z and T (more on these in the next section), we want to recover latent variables z_n, t_n for every sample x_n and learn two multi-class classifiers $f : \mathbb{R}^d \to \mathbb{R}^P$ and $g : \mathbb{R}^d \to \mathbb{R}^A$ (for persons and actions respectively). Because the two sets of classes correspond to very different aspects of the data, we define two feature maps ϕ and ψ that will be respectively taken as input for f and g. With a slight abuse of notations, we will represent with $\phi(X)$ (respectively $\psi(X)$) the matrix whose rows are the $\phi(x_n)$ (respectively $\psi(x_n)$).

2.2. Problem Formulation

Our problem can be decomposed as a sum of two cost functions (for person names and actions) that are linked by joint constraints. To avoid repetitions, we will first derive equations for one of the two cases only. Let us consider a multi-class loss function $\ell : \mathbb{R}^P \times \mathbb{R}^P \to \mathbb{R}$, and denote by $\Omega : F \to \mathbb{R}$ some regularization function over the set F of prediction functions under consideration. We formulate the recovery of the latent variables and the construction of the

classifier as the following optimization problem:

$$\min_{Z,f} \frac{1}{N} \sum_{i \in I} \sum_{n \in \mathcal{N}_i} \ell(z_n, f(\phi(x_n))) + \Omega(f) \qquad (1)$$

under some constraints defined in section 2.3. We define ℓ to be a square loss, the regularization term $\Omega(f)$ is defined by a squared $\mathcal{L}_2$ norm, and f is a linear classifier:

$$f(\phi(x_n)) = \phi(x_n)\, w + b\ ,\ w \in \mathbb{R}^{d \times P}\ ,\ b \in \mathbb{R}^{1 \times P}.$$

The optimization problem now becomes:

$$\min_{Z,w,b} \frac{1}{N} \|Z - \phi(X)w - b\|_F^2 + \lambda_1\, Tr(w^T\, w). \qquad (2)$$

Following [2], we note that (2) admits a closed form solution in w and b for fixed Z. Using this solution, we rewrite (2) as:

$$\min_{Z}\ Tr(ZZ^T A(X, \lambda_1)), \qquad (3)$$

where $A(X, \lambda_1)$ is a $N \times N$ matrix that depends on the data X and the regularization parameter λ_1.

Using (3), we next define a joint optimization problem over action labels T and person labels Z as:

$$\min_{Z,T}\ Tr(ZZ^T A(X, \lambda_1)) + Tr(TT^T B(X, \lambda_2)). \qquad (4)$$

Matrices A, B will be explicitly defined in Section 2.6.

Note that the above formulation does not contain any coupling between Z and T *per se*. We will use information mined from scripts to couple Z and T by joint constraints as described below.

2.3. Annotations as Constraints on Latent Variables

We would like to constrain solutions of our problem by coupling person and action labels. We do this using information mined from movie scripts. After aligning scripts with videos [9], we extract person-action pairs (p, a) and their approximate temporal locations. Given a pair (p, a) found in the script, we assume a person p performs an action a at the corresponding temporal location in the video. We model this assumption by constraints defined on the latent variables. To make the best use of the textual data, we distinguish three kinds of extracted pairs, (p, a), $(p, \varnothing)$ and $(\varnothing, a)$, leading to three types of constraints.

In scene descriptions found in scripts, we observe subject-verb pairs and associate those with either (p, a) or $(\varnothing, a)$ pairs. The distinction comes from the fact that some subjects may be pronouns and therefore not designate any specific character *a priori*.

The $(p, \varnothing)$ pairs come from another source of textual information: movie scripts contain both scene descriptions and dialogues with speaker identities specified. We therefore use this information to suggest speaker presence in the surrounding video.

For every person-action pair (p, a) we construct a bag i containing samples $\mathcal{N}_i$ corresponding to person tracks in the temporal proximity of (p, a). If multiple pairs have similar position in time, we group them, producing bags with several (p, a) pairs Λ_i. Once the bags are defined, we use annotations to constrain the latent variables of person tracks in the bag. What we want to model is the following: "if a person-action pair is mentioned in the script, it should appear at least once in the bag". This can be translated in the form of constraints on sums of latent variables of tracks within a bag as:

$$\begin{aligned} \forall i \in I\ ,\ \forall (p, a) \in \Lambda_i\ ,\ \sum_{n \in \mathcal{N}_i} z_{np}\, t_{na} \geq 1, \qquad (5) \\ \forall (p, \varnothing) \in \Lambda_i\ ,\ \sum_{n \in \mathcal{N}_i} z_{np} \geq 1, \\ \forall (\varnothing, a) \in \Lambda_i\ ,\ \sum_{n \in \mathcal{N}_i} t_{na} \geq 1. \end{aligned}$$

Constraints based on (p, a) provide coupling between the two sub-parts of our problem. Pairs $(p, \varnothing)$ and $(\varnothing, a)$ define independent constraints on the person and action latent classes respectively.

Since we have partial knowledge about class membership of samples in bags, our problem is related to multiple instance learning (MIL) [24]. MIL, however, is not clearly defined for the multi-class case. In the binary case it consists in learning a binary classifier given bags containing samples of both classes and bags containing only negatives. When considering an analogous problem in the multi-class case, it is unclear what the bag assumptions would be.

2.4. Slack Variables

In practice, person-action pairs in scripts may not always have corresponding person tracks in the video. This can happen due to failures of automatic person detection and tracking as well as due to possible mismatches between scripts and video tracks. To cope with these issues, we introduce slack variables allowing the constraints to be violated. We define a vector ξ of length $\sum_{i \in I} |\Lambda_i|$ and rewrite our problem as:

$$\begin{aligned} &\min_{Z,T,\xi}\ Tr(ZZ^T A(X, \lambda_1)) + Tr(TT^T B(X, \lambda_2)) + \kappa\, \xi^T \xi \\ &\text{s.t. } \forall i \in I\ ,\ \forall\, J \in \Lambda_i\ : \\ &\quad \begin{cases} \sum_{n \in \mathcal{N}_i} z_{np}\, t_{na} \geq 1 - \xi_J & \text{if } J = (p, a), \\ \sum_{n \in \mathcal{N}_i} z_{np} \geq 1 - \xi_J & \text{if } J = (p, \varnothing), \\ \sum_{n \in \mathcal{N}_i} t_{na} \geq 1 - \xi_J & \text{if } J = (\varnothing, a). \end{cases} \qquad (6) \end{aligned}$$

2.5. Optimization

When Z and T take binary values, solving the problem defined by eq. (6) is NP hard. We thus relax it by considering real-valued positive matrices Z, T such that $Z\,\mathbf{1}_P = 1$, $Z \geq 0$ and $T\,\mathbf{1}_A = 1, T \geq 0$.

The relaxed problem is not jointly convex in Z and T because of the coupling constraint in eq. (5). Once we fix one of the two matrices, the coupling constraint becomes linear in the other latent variable. We, therefore, perform a block coordinate descent and alternate optimization by solving for one of the matrices Z, T while fixing the other. Each of the two steps is a convex quadratic program under linear constraints since A and B are positive-semidefinite by construction.

In the first step we freeze the T variable and optimize over Z and ξ. We initialize T with the uniform assignment matrix $T = \frac{1}{A}\mathbf{1}_N\mathbf{1}_A^T$. Since the two steps are separately convex, the initialization of Z does not matter.

Rounding. Given estimates of real-valued matrices Z and T, we have to choose classes for every sample. To do so we compute the orthogonal projection according to $\mathcal{L}_2$ norm on the set of indicator matrices $\mathcal{Z} = \{Z \in \{0,1\}^{N\times P} \mid Z\mathbf{1}_P = \mathbf{1}_N\}$:

$$\arg\min_{\hat{Z}\in\mathcal{Z}} \|\hat{Z} - Z\|_2. \tag{7}$$

This amounts to taking maximum values along rows of Z and T. For each row the arguments of the maximum define classes of corresponding samples while the maximum values are used as confidence values in our evaluation.

Relation to Diffrac [2]. Our problem formulation in (4) is closely related to the discriminative clustering approach Diffrac [2, 17]. When latent classes are treated equally, the minimization of a convex relaxation of (4) results in a trivial solution [12]. To overcome this issue one can perform a lifting and optimize (4) with respect to the equivalence matrix $M = ZZ^T$ instead (under a suitable set of constraints).

Working with M is problematic in our case since our constraints in (5) are defined on the elements of Z rather than on M. Class-dependent constraints in our case, however, break the symmetry in class labels and enable (4) to be solved directly for Z. In practice we found that modifying the value of 1 to a larger constant on the right sides of inequalities (5) leads to a more stable solution of (4).

2.6. Use of Kernels

As mentioned in [2], the optimization problem (3) allows the use of kernels. Using the matrix inversion lemma, one can derive an expression for A and B that depends only on the Gram matrix of the linear kernel (XX^T). We can, therefore, replace XX^T by the Gram matrix of any kernel, yielding in our case:

$$\begin{cases} A(K_1,\lambda_1) & = \lambda_1\Pi_N\left(\Pi_N K_1 \Pi_N + N\lambda_1 I_N\right)^{-1}\Pi_N, \\ B(K_2,\lambda_2) & = \lambda_2\Pi_N\left(\Pi_N K_2 \Pi_N + N\lambda_2 I_N\right)^{-1}\Pi_N, \end{cases}$$

where Π_N is the projection matrix $\Pi_N = I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ and

$$\forall (i,j) \in \{1,\ldots,N\}^2\,,\, (K_1)_{i,j} = K_f(\phi(x_i),\phi(x_j)),$$
$$(K_2)_{i,j} = K_a(\psi(x_i),\psi(x_j)).$$

K_f and K_a are the two kernels that we use for faces and actions as described in more details in Section 3.

3. Features and Dataset

In this section we describe features extracted from the text and the video, and give details about the used dataset.

Text processing. We extract person-action pairs from text using a semantic role labeling parser. Semantic role labeling consists of identifying arguments (agent, instrument, manner, cause) to a predicate (for example a verb). Intuitively, this amounts to answering questions such as "Who" "What" "When" "Where" "Why". Several statistical parsers are available on-line. We use SEMAFOR [7], which is trained on the FrameNet database [3]. We focus on two frames that appear often enough in the script and have an associated agent: "ChangePosture" and "SelfMotion". From each detected occurrence of the frame in the text we use the "agent" and the "target verb" as the name and action pair.

Video features. The aim here is to design a representation of video that can be related to the name and action structures extracted from the text. This is achieved by automatically extracting tracks of people from video. Each person is then represented by its face appearance to capture identify and motion features to represent the action. See figure 2.

To extract person tracks, we run the multi-view face detector of [26] and associate detections across frames using point tracks in a similar manner to [9, 22]. To represent faces we follow [22], and extract facial features and rectify each face into a canonical frame using a similarity transformation. We then re-compute facial feature positions in the rectified image and extract SIFT descriptors at multiple scales from each facial feature. The descriptor for each face is formed by the concatenation of all SIFT descriptors. Finally, each track is represented by the set of descriptors, one for each face in the track.

To represent actions, we compute bag-of-features on dense trajectories [25] extracted from each person track. We take the trajectories that fall into the spatio-temporal volume defined by the upper-body bounding box in each frame. The

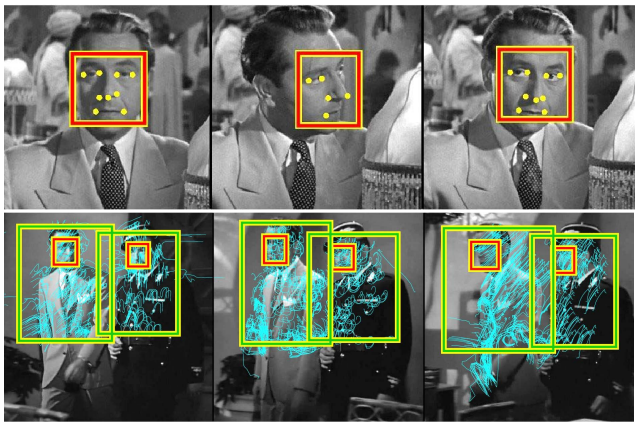


Figure 2: **Representing video.** Top: face track together with extracted facial features. Bottom: Motion features based on dense point trajectories extracted from tracked upper body bounding boxes.

upper-body bounding box is defined here by simply extrapolating the face bounding-box using a linear transformation. This assures that in every frame we have a corresponding face as well as an upper-body region. Our discriminative cost function allows the use of kernels. For face tracks, we follow [22] and use the sum of "min-min kernels" computed separately for each facial feature as well as frontal and profile faces. This results in a total of 38 face track kernels (24 for frontal features and 14 for profile features) that are summed with uniform weights. For action descriptors we use the exponentiated chi-square kernel [25].

Dataset. We report results for movies Casablanca and American Beauty. For both movies we extract person tracks and associated descriptors. We discard person tracks with unreliable facial features based on the landmark localization score. For Casablanca, we obtain 1,273 person tracks containing 124,423 face detections while for American Beauty we use 1,330 person tracks containing 131,741 face detections.

By processing corresponding movie scripts, we extract 17 names for the main characters in Casablanca and 11 names for the main characters in American Beauty. For each movie we select two most frequent action classes, i.e., *walking, sit down* for Casablanca and *walking, open door* for American Beauty. For Casablanca we obtain 42 action/name pairs and 359 occurrences of names with no associated actions. For American Beauty the corresponding numbers are 31 and 330, respectively. The dataset is available at [1].

To explicitly model non-named characters in the movie (side characters and extras) as well as non-considered action classes we introduce an additional "background" class for both faces and actions. We collect background examples as follows. For faces, we collect additional 300 random faces from the Labeled Faces In The Wild dataset [15]. For actions, we randomly sample 500 person tracks from the Hollywood2 dataset [20] using the corresponding movie scripts to discard actions considered in this work. For all "background" samples, we constrain latent variables to take values corresponding to the "background" class. We found that including this additional data helps resolving confusion in label assignment for our target classes.

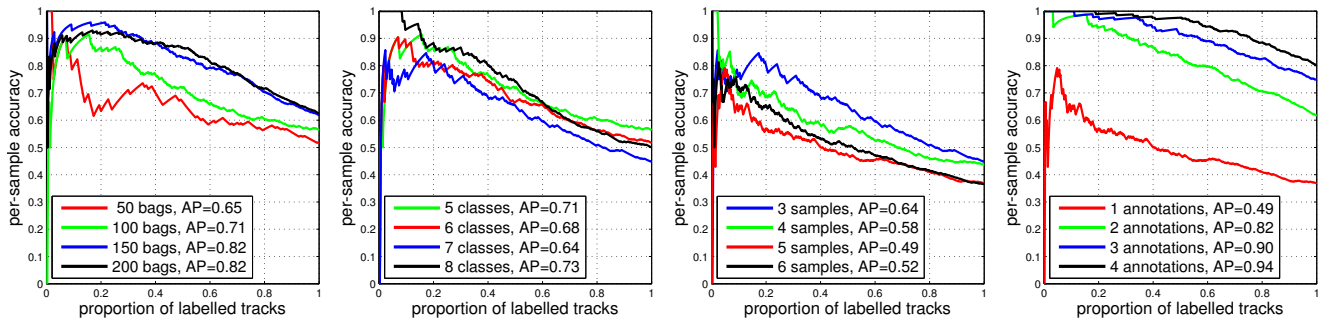
4. Experiments

In this section we experimentally demonstrate the benefits of the proposed approach. We first test the sensitivity to parameter choices in a controlled character identification setup. Second, we show that even for learning names alone (without actions) the proposed method outperforms other state-of-the-art weakly supervised learning techniques designed for the same task. Finally, we demonstrate benefits of learning names and actions jointly compared to resolving both tasks independently.

Learning names: controlled set-up. Here we wish to assess the sensitivity of the proposed method to the following four important parameters: the number of bags $|I|$, the number of classes P, the number of samples per bag $|\mathcal{N}_i|$ and the number of annotations per bag $|\Lambda_i|$. We will use real data – 1,273 face tracks and their descriptors from the movie Casablanca – but group the tracks into bags in a controlled manner. Each track is labeled with a ground truth name from the set of 18 main characters (or other). To create each bag, we first sample a track from a uniform distribution over characters and then complete the bag with up to $|\mathcal{N}_i|$ tracks by randomly sampling tracks according to the true distribution of the characters in the movie. Each bag is annotated according to the first sample. Given this data, we solve the sub-problem related to faces, i.e. no joint action labels are used in this experiment.

As discussed in Section 2, each face track is assigned to a class by maximizing the rows of Z. Classified face tracks are then sorted by their confidence values and the percentage of correctly classified tracks (i.e., the per-sample accuracy) is evaluated for each confidence value. Following [9, 22] we measure performance by plotting a curve of per-sample accuracy vs. proportion of labeled tracks. Ideally, the accuracy would be one for all confidence values, but in practice the accuracy drops for samples with lower confidence. We illustrate results for different bag layouts in Figure 3.

Comparison with other weakly supervised methods. Here we compare our method with other weakly supervised face identification approaches. We use the code adapted from [22] and an on-line available implementation of [6].



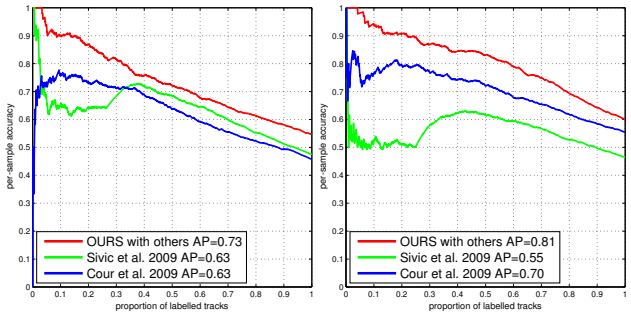
(a) (b) (c) (d)

Figure 3: **Performance for different bag layouts in a controlled set-up. (a)** First, we vary the number of bags while fixing 3 samples and 1 annotation per bag, and the number of classes to 5. As expected, performance improves with more bags. **(b)** Keeping 150 bags in total, we increase the number of classes. The effects of this modification are mixed. By adding more classes, the problem is harder but the per bag confusion is smaller. **(c)** Keeping 7 classes, we increase the number of samples per bag showing that more samples per bag increase confusion resulting in a lower performance. **(d)** Keeping 5 samples per bag, we increase the number of annotations per bag, clearly showing the benefits of having more annotations.

We run all methods on 1,273 face tracks from Casablanca and 1330 face tracks from American Beauty using noisy name annotations obtained from movie scripts. To have a fair comparison, no action labels are used. While [6] and [22] have been evaluated on television series, here we address a more challenging setup of full-length movies. First, the training data within a film is limited as it is not possible to harvest face tracks across multiple episodes as in TV series. Second, the cast of characters in a film is often larger than in TV series with many additional extras. Third, films often employ a wider set of cinematographic techniques compared to often simpler structure of a TV show with many close-ups and "shot-reverse shot" dialogues.

Comparative results for the two movies in Figure 4 demonstrate superior performance of our method. The lower performance of [22] can be explained by its dependency on the visual speaker identification. While our adaptation of the code obtained from the authors of [22] worked well on their data, we found that the speaker detection achieved only 64.2% and 50.2% accuracy (with about 25% speaker labeled tracks) on Casablanca and American Beauty, respectively. The lower accuracy, compared to the accuracy of more than 80% on the TV series data from [22], could be possibly due to the challenging illumination conditions with strong shadows present in the two films. The approach of [6] assumes that correct labels are included into the set of "ambiguous" labels. This assumption is often violated in movies as side characters and extras are often not mentioned in the script. In contrast, our approach suffers less from this problem since (a) it can handle multiple annotations for bags of multiple tracks and (b) the noise in labels and person detections is explicitly modeled using slack variables.

Learning names and actions. We next evaluate benefits of learning names and actions jointly. This is achieved by



(a) Casablanca (b) American Beauty

Figure 4: Results of automatic person naming in movies. Our method is compared with weakly supervised face identification approaches of Cour *et al.* [6] and Sivic *et al.* [22].

first learning the name assignments Z for all tracks. The name assignments are then fixed and used as additional constraints when learning the likely action assignments T for each track. While this procedure can be iterated to improve the assignment of actor names with the help of estimated action labels, we found that the optimization converges after the first iteration in our setup.

The distribution of action classes in our data is heavily unbalanced with the "background" class corresponding to more than 78% of person tracks. We therefore evaluate the labeling of each target action in each movie using a standard one-vs-all action precision-recall measure. We compare the following methods. **Names+Actions** corresponds to our proposed method of learning person names and actions jointly. **No Names** uses constraints on actions only without considering joint constraints on actions and names. **True Names+Actions** uses the ground truth person names as constraints on actions instead of the automatic name assignment. This provides an upper bound on the action classification performance provided perfect assignment of per-

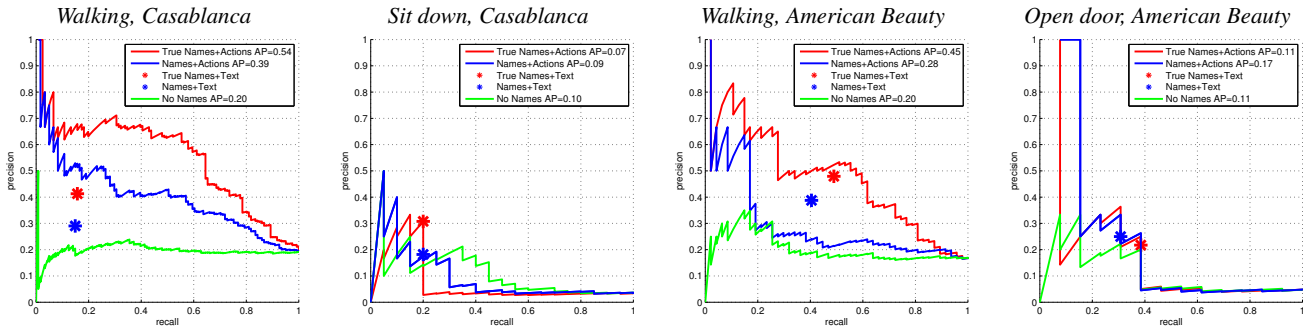


Figure 5: Results of action labeling in movies Casablanca and American Beauty. See Section 4 for more details.

son names. Finally, we evaluate two "dummy" baselines which blindly assign action labels based on person names and person-action pairs obtained from scripts. The purpose of these baselines is to verify that visual action classification improves the performance. **Names+Text** learns face assignments for each person track and assigns action labels using person-action pairs. **True Names+Text** assigns action labels based on person-action pairs and ground truth person names. This baseline, hence, does not "look" at image pixels at all. Note that the last two baselines produce a single point on the precision-recall plot as no confidence values are available when transferring action labels from scripts.

Precision-recall plots for the target action classes in two movies are shown in Figure 5. We first observe that our full method (blue curves) outperforms the weakly supervised learning of actions only (green curves) in most of the cases. This shows the benefit of learning actions and names jointly. As expected, action classification can be further improved using ground truth for name assignments (red curves).

For the frequent action *walking* for which many person-action constraints are available in scripts, automatic person naming in our method provides a large benefit. However, even with ground truth face assignments the action classification performance is not perfect (True Names+Actions). This is likely due to two reasons. First, the ambiguity in the weak supervision is not reduced to zero as a single character may do several different actions in a single clip (bag). Second, the current action representation has only limited discrimination capabilities.

Recognizing less frequent actions *sit down* and *open door* appears to be more difficult. While several examples are ranked high, all methods suffer from a small number of available person-action constraints. In addition, a significant portion of these constraints is incorrect. Incorrect constraints often occur due to the failure of face detection as actors often turn away from the camera when sitting down and opening doors. To explicitly quantify the loss due to failures of automatic person tracking, we have manually annotated person tracks in the movie Casablanca. The performance of our full method is significantly improved when run on correct person tracks yielding AP=0.36 and AP=0.63 for the *sit down* and walk actions, respectively. This emphasizes the need for better automatic person detection and tracking methods. Qualitative results for automatic labeling names of actors and actions using our method (Names+Actions) are illustrated Figure 6. More results are available at [1].

5. Conclusion

We have developed a new discriminative weakly supervised model jointly representing actions and actors in video. We have demonstrated the model can be learnt from a feature length movie together with its shooting script, and have shown a significant improvement over other state-of-the-art weakly supervised methods. As actions are shared across movies, applying the model over multiple movies simultaneously opens-up the possibility of automatically learning discriminative classifiers for a large vocabulary of action classes.

Acknowledgements This research was supported in part by the ERC advanced grant VideoWorld, the ERC starting grant Activia, the ERC advanced grant Allegro, Quaero project, the MSR-INRIA laboratory, the European integrated project AXES, Google and the Institut Universitaire de France.

References

[1] http://www.di.ens.fr/willow/research/actoraction, 2013. 5, 7, 8

[2] F. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In *NIPS*, 2007. 2, 3, 4

[3] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *COLING-ACL*, 1998. 4

[4] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *J. Machine Learning Research*, 2003. 2

[5] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. G. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *CVPR*, 2004. 1, 2

[6] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *CVPR*, 2009. 1, 2, 5, 6

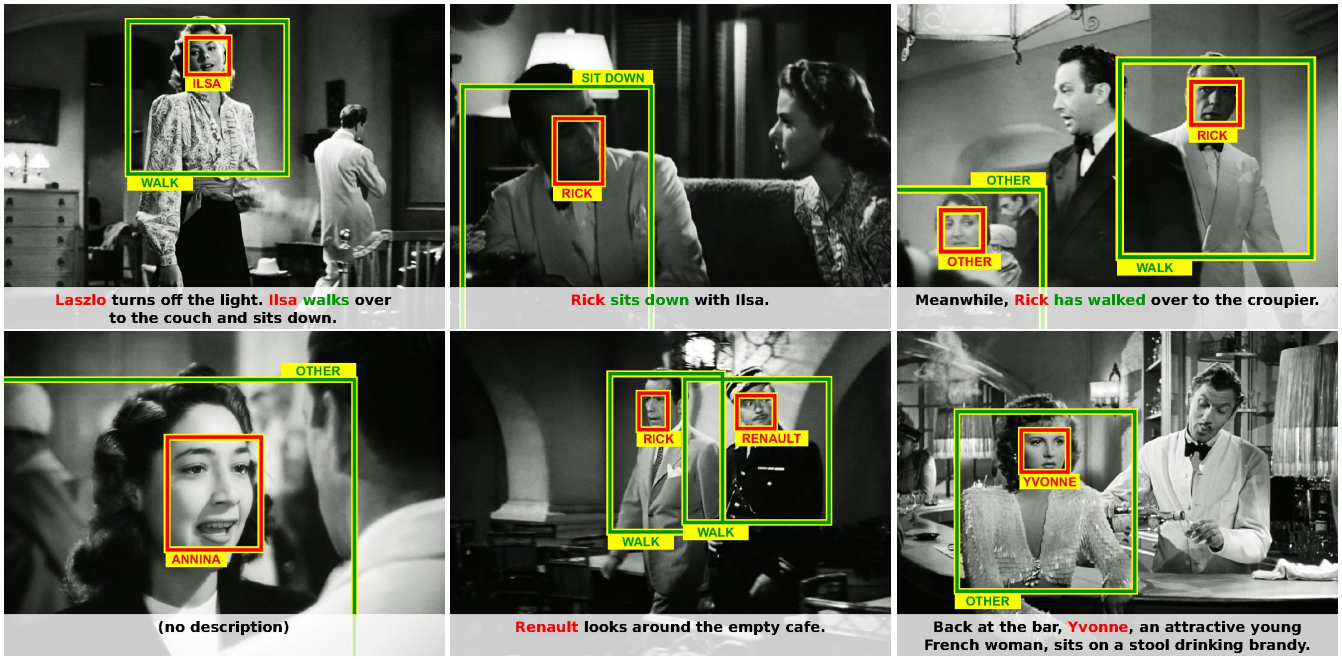


Figure 6: **Examples of automatically assigned names and actions in the movie Casablanca. Top row:** Correct name and action assignments for tracks that have an actor/action constraint in the script. **Bottom row:** Correct name and action assignments for tracks that do not have a corresponding constraint in the script, but are still correctly classified. Note that even very infrequent characters are correctly classified (Annina and Yvonne). See more examples on the project web-page [1].

[7] D. Das, A. F. T. Martins, and N. A. Smith. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *SEM*, 2012. 4

[8] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 1, 2

[9] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *BMVC*, 2006. 2, 3, 4, 5

[10] A. Farhadi, M. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: generating sentences for images. In *ECCV*, 2010. 2

[11] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *CVPR*, 2009. 2

[12] Y. Guo and D. Schuurmans. Convex relaxations of latent variable training. In *NIPS*, 2007. 4

[13] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008. 1, 2

[14] A. Gupta, P. Srinivasan, J. Shi, and L. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009. 2

[15] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 2007. 5

[16] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 2

[17] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 2, 4

[18] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 1, 2

[19] J. Luo, B. Caputo, and V. Ferrari. Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In *NIPS*, 2009. 1, 2

[20] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 1, 5

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

[22] J. Sivic, M. Everingham, and A. Zisserman. "who are you?" - learning person specific classifiers from video. In *CVPR*, 2009. 1, 2, 4, 5, 6

[23] M. Tapaswi, M. Bauml, and R. Stiefelhagen. "knock! knock! who is it?" probabilistic person identification in tv-series. In *CVPR*, 2012. 1

[24] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization. In *CVPR*, 2008. 3

[25] Y. Wang and G. Mori. A discriminative latent model of image region and object tag correspondence. In *NIPS*, 2010. 2, 4, 5

[26] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 4