Script Data for Attribute-Based Recognition of Composite Activities

Marcus Rohrbach[1], Michaela Regneri[2], Mykhaylo Andriluka[1], Sikandar Amin[1,3], Manfred Pinkal[2], and Bernt Schiele[1]

[1] Max Planck Institute for Informatics, Saarbrücken, Germany
[2] Department of Computational Linguistics, Saarland University, Germany
[3] Department of Computer Science, Technische Universität München, Germany

Abstract. State-of-the-art human activity recognition methods build on discriminative learning which requires a representative training set for good performance. This leads to scalability issues for the recognition of large sets of highly diverse activities. In this paper we leverage the fact that many human activities are compositional and that the essential components of the activities can be obtained from textual descriptions or scripts. To share and transfer knowledge between composite activities we model them by a common set of attributes corresponding to basic actions and object participants. This attribute representation allows to incorporate script data that delivers new variations of a composite activity or even to unseen composite activities. In our experiments on 41 composite cooking tasks, we found that script data to successfully capture the high variability of composite activities. We show improvements in a supervised case where training data for all composite cooking tasks is available, but we are also able to recognize unseen composites by just using script data and without any manual video annotation.

1 Introduction

Human activity recognition in video is a fundamental problem in computer vision. State-of-the-art methods (e.g. [1–3]) achieve near perfect results for simple actions (e.g. KTH dataset [4]), and robustly recognize actions in realistic settings such as Hollywood movies [5], videos from YouTube [6], or sport scenes [7].

The top-performing methods typically rely on discriminative machine learning, which requires representative training data. Collecting such training sets is challenging if the number of activities is large and the activities themselves are complex. In consequence, most current research (with few exceptions [8, 3, 9]) focuses on simple basic-level activities such as *walking* or *drinking*, while the recognition of longer-term, complex, and composite activities such as *assembling furniture* or *food preparation* has been rarely addressed in computer vision.

A promising approach towards scalability of activity recognition methods to a large number of complex activities is to use intermediate representations that are shared and transferred across activities by exploiting their compositional nature. We exploit this technique and propose a new approach building on an attribute-based representation. Instead of learning a model for each composite activity we

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part I, LNCS 7572, pp. 144–157, 2012.

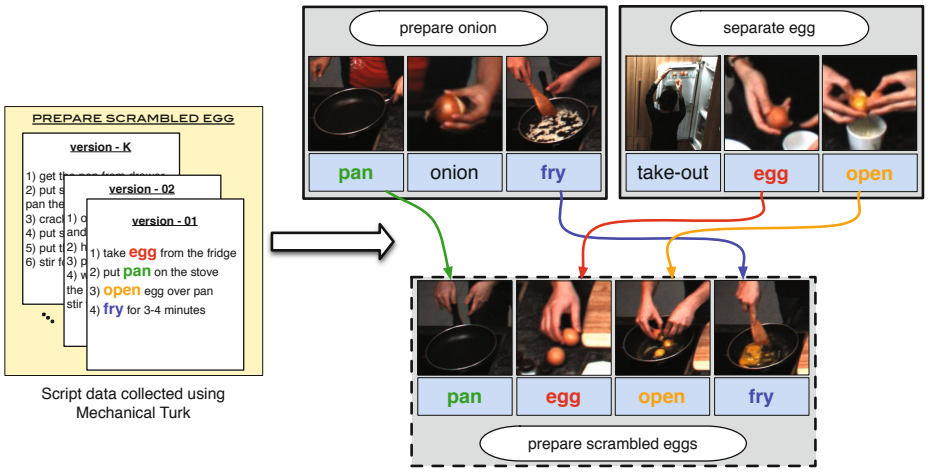


Fig. 1. Sharing or transferring attributes of composite activities using script data. Composite activities (gray boxes) are composed of basic-level activities and their participants (light-blue boxes), modeled as attributes. These attributes can be transferred with the help of script data to unseen composite activities (dashed-line box).

learn models for a large set of attributes shared across composite activity classes. Such approaches have been shown effective to recognize previously unseen object categories [10, 11] and have also been applied to activity recognition [12].

We evaluate our approach in the daily living domain where many tasks, such as *cleaning the house* or *preparing a dish*, are composed of several basic-level activities. A major challenge to recognize everyday activities is that these activities can often be performed in a wide variety of ways, and it is practically infeasible to create a training set with all possible alternatives.

For the purpose of this paper we focus on the recognition of cooking activities, which share many basic-level activities, cooking tools, and ingredients. A recent evaluation [13] has shown that recognizing basic-level activities is already a challenging task and thus the recognition approach needs to be robust to failures in detection of basic-level activities. In this work we address the challenges of difficult basic-level cooking activities as well as the high variability in composite activities in three complementary ways:

1. We detect activities and objects independently but take their co-occurrence and context into account. E.g. when looking at cooking activities it is likely that *peeling* co-occurs with *carrot* or *potato* but not with *cauliflower*.
2. We model basic-level activities and participants as attributes of composite activities, allowing to easily share and transfer across composite activities. As Fig. 1 shows a decomposition of the activities *prepare onion*, *separate egg*, and *prepare scrambled eggs* into attributes of basic-level activities such as *fry* and *open* as well as their participating ingredients (*egg*) or tools (*pan*).
3. We collect a large number of textual descriptions, instances of so called scripts, for an activity to compute how relevant a certain attribute is for a

specific composite activity. Given this script data we can not only handle the variation of composites but also recognize unseen composite activities. As illustrated in Fig. 1 the attributes *egg*, *pan*, *open*, and *fry* are determined to be important for *preparing scrambled eggs* using script data and can be transferred from known composites such as *separate egg* and *prepare onion*.

Our contributions are as follows. First, we show how to use text-based script data for handling the large variability of composite activity recognition by selecting relevant attributes. Second, we not only improve performance in the supervised case but also can transfer to unseen composite cooking activities. We achieve this by decomposing composite activities into a flexible attribute representation. Third, we show that using co-occurrence and temporal activity context can help recognizing the challenging basic-level activities. Additionally, we release the challenging recorded video dataset (called *MPII Cooking Composite Activities*, or short *MPII Composites*) allowing to evaluate recognition of activity composites and attribute transfer on a large scale.

2 Related Work

This paper addresses the challenging task to recognize complex everyday activities, taking cooking as running example. Our goal is to leverage on the compositional nature of human activities to enable the recognition of activities for which only few or even no training examples are available. This is in contrast to approaches that represent activities as bags of spatio-temporal features [14, 1, 2, 15] disregarding potential structure within the activity.

Several recent approaches [16, 3, 12] have aimed at structured representation of activities that go beyond bags-of-features. Joint modeling of actions and objects has been explored in [17, 16], demonstrating improved performance for both tasks. In this work we also include both actions and objects in our representation, while aiming to recognize more complex interactions and activities. [3] model activities as a temporal composition of primitive actions and discriminatively learn such models. The primitive actions are learned in a data-driven manner complicating transfer to previously unseen activities. In contrast to this we focus on semantically meaningful basic-level activities, which permit to learn the relationships between activities and objects from textual sources.

Recent work has shown that attributes are an effective intermediate representation that facilitates cross-task [10] and cross-modal learning [11]. We build our approach on such an representation using attributes that are commonly shared between cooking activities. The attributes correspond either to basic-level activities such as *stir*, *peel*, or *grate* or to tools and ingredients used in the cooking process. Our representation is conceptually similar to a recently proposed *object/action bank* representation for scene recognition [18], for still image action recognition [19], and video action recognition [19]. Similar to these, we first train a set of detectors for a large set of attributes and then perform reasoning on top of the detector bank output.

While attributes have been used for object recognition [20, 10, 21, 11] they have only recently been applied to activity recognition [19, 12]. [12] builds on a set of manually defined attributes describing various body motions such as *raise arms* and *bend torso*. The attributes are interpreted as latent variables and combined with motion trajectory features and attribute co-occurrence features within a latent SVM framework. [12] demonstrates the effectiveness to recognize previously unseen activities, but requires manual specification of activity attributes. In contrast to this we put our main focus on investigating how attribute relationships can be automatically mined from text sources.

Language and cross-modal learning have been used for knowledge transfer [17, 11]. In [17] visual and RFID data are combined with common-sense knowledge to learn recognition models of complex kitchen activities. [11] relied on publicly available databases such as Wikipedia[1], WordNet [22], or Flickr[2] to mine relationship between attributes and objects, and uses them to recognize novel object classes. Methods such as [11] have not been explored for activity recognition in the past, likely because generic text corpora do not seem suitable for mining activity-related attributes as noted by [12]. To address this, we explicitly gather knowledge about activities by collecting their textual descriptions from multiple subjects. We then rely on linguistic analysis of such descriptions in order to compute statistics of the appearance of various attributes within each activity. We demonstrate that such statistics allow to significantly boost recognition performance and also facilitate recognition of previously unseen activities.

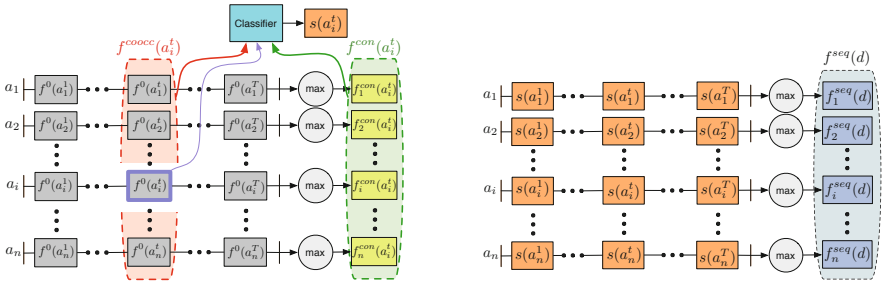
Movie scripts associated to a movie have previously been used by [23] to obtain automatic annotations of activities, in contrast to this we want to capture unseen variations by script data collected independent of the video. In the multimedia community, MediaMill[24] and LSCOM[25] are efforts to explore large scale video retrieval using mid-level concepts and exploring combination of textual and visual information.

3 Modeling Attributes and Composite Activities

We are interested in two activity recognition tasks: First we would like to recognize different composite activities, such as preparing cucumbers. Secondly, we want to recognize the various activity attributes associated to and making up the composite activity. Those attributes characterize the composite activity and are either basic-level activities (such as *peeling* or *washing*) or the respective participants (such as *grater*, *knife*, or *cucumber*).

This section first describes the attribute recognition approach that equally applies to basic-level activities and participants (Sec. 3.1). Composite activities are recognized based on these attributes (Sec. 3.2). We then show how to use prior knowledge (Sec. 3.3) to improve the recognition of composite activities, overcoming the notorious lack of training data and handling the large variability of activities.

[1] `http://www.wikipedia.org`
[2] `http://www.flickr.com`



(a) Activity attribute recognition using contextual and co-occurrence attributes

(b) Composite activity classification using activity attributes

Fig. 2. Our approach to recognition of attributes (a) and composite activities (b)

3.1 Recognizing Activity Attributes Using Context and Co-occurrence

For a time interval t we want to classify if a particular activity attribute a_i is present. As mentioned before a_i can be any attribute including *cut*, *knife*, or *cucumber*. To obtain the final classifier score for an attribute a_i we are proposing to use three different types of features. The first type of feature is given by a video-feature-based attribute classifier providing us with confidence score $f^0(a_i^t)$ for attributes a_i at time interval t. In addition to $f^0(a_i^t)$, we define features based on context (in the same video sequence) as well as features based on the co-occurrence of other attributes (in the same time interval t).

Contextual features formalize the intuition that close or adjacent time frames have strongly related attributes: E.g. if a *cucumber* is *peeled* in one time interval, the *cucumber* is probably also present in the surrounding time frames, and it is likely that the same video sequence contains a *cutting* activity as well. More formally (visualized in Fig. 2(a)) we define a context feature vector $f^{con}(a_i^t)$ as

$$f_j^{con}(a_i^t) = \max_{u=1,\ldots,t-1,t+1,\ldots,T} f^0(a_j^u) \qquad \forall j \in \{1,2,\ldots n\}, \tag{1}$$

where n is the total number of attributes. Element j of the context feature vector contains evidence that attribute a_j occurs in the context of attribute a_i.

Similarly, activity attributes happening at the same time instance t are related, e.g. if we *peel* something it is more likely to observe also *carrot* or *cucumber* rather than *cauliflower*. We thus define the co-occurrence by a feature vector $f^{coocc}(a_i^t)$ of all attribute scores excluding a_i^t:

$$f_k^{coocc}(a_i^t) = f^0(a_k^t) \qquad \forall k \in \{1\ldots n\}\backslash i \tag{2}$$

Based on these features we train an activity attribute classifier using the features individually or by stacking them (see Fig. 2(a)). This formulation can be easily extended to other attribute representations depending on the task and available features. In the following, $s(a_i)$ refers to the score of such an attribute classifier.

3.2 Composite Activity Classification Using Activity Attributes

We now want to classify composite activities that span an entire video sequence, given attribute classifier scores $s(a_i^t)$. In this approach we rely on the representation that captures likelihoods of the presence or absence of a particular attribute and leave modeling temporal ordering of attributes for future work. For each sequence d we build a feature vector $f^{seq}(d)$ by computing the maximum score of each attribute over all time intervals (see Fig. 2(b)):

$$f_i^{seq}(d) = \max_{t=1,\dots,T} \{s(a_i^t)\} \qquad \forall i \in \{1, 2, \dots n\}. \tag{3}$$

To decide on the category of a sequence we use the feature representation $f_i^{seq}(d)$ and classify using a nearest neighbor classifier (NN) or support vector machines (SVM) given a set of labeled training sequences. The following section describes the additional incorporation of semantic relatedness to select the relevant attributes a_i, i.e. feature dimensions in $f_i^{seq}(d)$.

3.3 Script Data for Recognizing Composite Activities

Composite activities show a high diversity which is practically impossible to capture in a training corpus. Our system thus needs to be robust against many activity variants that are not present in the training data. The use of attributes allows to include external knowledge to determine relevant attributes for a given composite activity. For this we assume associations between attribute a_i and composite activity class z in a matrix of (normalized) weights w_i^z. Our system extracts those associations from script data (see Sec. 4), but the approach generalizes to arbitrary other external knowledge sources. We explore two options to use such information, one of which does not require any visual training data of a specific composite activity and thus enables zero-shot recognition.

Script Data: To compute a confidence score $s^{scriptdata}(z|d)$ of the composite activity d being of class z we use the attribute based feature representation $f_i^{seq}(d)$. Given the weights w_i^z we compute a weighted sum

$$s^{scriptdata}(z|d) = \frac{\sum_{i=1}^{n} w_i^z f_i^{seq}(d)}{\sum_{i=1}^{n} w_i^z}, \tag{4}$$

This formulation is similar to the sum formulation by [26] used for image recognition with attributes, which itself is an adaption of the direct attribute prediction model introduced by [10]. Note that the weight matrix retrieved from script data is sparse (often, $w_i^z = 0$). When mining from other corpora one might need to threshold or cut-off the weights w_i^z to achieve good performance.

NN+script Data: When training data is available we can use a nearest neighbor classifier. Often, only a handful of attributes are likely to be indicative for a composite activity class, while the majority of other attributes will provide irrelevant, potentially noisy information. When searching for nearest neighbors

such irrelevant attributes might dominate the distance, resulting in suboptimal performance. To reduce this effect we rely on the script data to constrain the attribute feature vector to the relevant dimensions.

More specifically, we replace the distance measure of nearest neighbor with the following training class dependent similarity function, taking weights of class-attribute associations into account. It is defined between the test attribute vector of unknown class $f_i^{seq}(d^{test})$ and the training attribute vector $f_i^{seq}(d_z^{train})$ of class z as

$$Sim(d^{test}, d_z^{train}) = \left(\frac{\sum_{i=1}^{n} w_i^z \left(f_i^{seq}(d^{test}) - f_i^{seq}(d_z^{train}) \right)^2}{\sum_{i=1}^{n} w_i^z} \right)^{0.5} . \quad (5)$$

To enhance robustness further, we binarize all association weights w_i^z by setting all non-zero weights to 1. This reduces the distance computation to the relevant attributes, normalized by the total number of relevant attributes. Using continues weights requires their inversion, which performed worse than binarized weights for our purposes.

4 Mining Script Data

Linguistics and psychology literature knows prototypical sequences of certain activities as so-called *scripts* [27, 28]. Scripts describe a certain scenario (e.g. "eating in a restaurant") with temporally ordered events (*the patron enters restaurant, he takes a seat, he reads the menu...*) and participants (*patron, waiter, food, menu,...*). Written event sequences for a scenario can be collected on a large scale using crowdsourcing [29]. We make use of this method regarding our composite activities as scenarios and assembling a large number of written sequences for each of those. After a more detailed description of the data collection, we show how to match attribute labels to the text data, and what kind of statistics we use to compute the association weights w_i^z in Eq. 4 and 5.

4.1 Data Acquisition via Crowdsourcing

We collect natural language sequences similar to [29] using Amazon's Mechanical Turk[3]. For each composite activity, we asked the subjects to give tutorial-like sequential instructions for executing the respective kitchen task. The instructions had to be divided into sequential steps with at most 15 steps per sequence. We select 53 relevant kitchen tasks as composite activities by mining the tutorials for basic kitchen tasks on the webpage "Jamie's Home Cooking Skills"[4]. All those tasks are steps to process ingredients or to use certain kitchen tools. In addition to the data we collected in this experiment, we use data from the OMICS corpus[5] and [29] for 6 kitchen-related composite activities. This results in a corpus with 2124 sequences in sum, having a total of 12958 event descriptions.

[3] `http://www.mturk.com`
[4] `http://www.jamieshomecookingskills.com`
[5] `http://openmind.hri-us.com/`

1. get a large sharp knife	1. gather your cutting board and knife.	1. wash the cucumber
2. get a cutting board	2. wash the cucumber.	2. peel the cucumber
3. put the cucumber on the board	3. place the cucumber flat on the cutting board.	3. place cucumber on a cuttingboard.
4. hold the cucumber in your weak hand	4. slice the cucumber horizontally into round slices.	4. take a knife and rock it back and forth on the cucumber
5. chop it into slices with your strong hand		5. make a clean thin slice each time.

Fig. 3. 3 example scripts for the composite activity *cutting a cucumber*

This dataset provides much more variation than the limited number of video training examples can capture. Of course this poses also a challenge, because we need to overcome the problem of different wordings and coordinated events: Fig. 3 shows three examples we collected for the composite activity *chopping a cucumber*. They differ in verbalization (cf. *slice*, *chop* and *make a slice*) and granularity (*getting* something is often left out). Further, the sequences reflect different ways of preparing the vegetable, some include *peeling* it, some do not *wash* it, and so on. Some sentences contain conjugated events (*take a knife and rock it...*). While we clean the data to a certain degree by fixing spelling mistakes and resolving pronouns with the method in [30], we end up with both challenges and blessings of a very noisy but very big training data set.

4.2 Data Analysis

To use the prior knowledge from the textual data, we match the attribute labels from the video annotations to the written script instances and compute several statistics: the frequency distribution give simple priors of single attributes, and tf*idf is used to find the most salient composite activity associated with certain basic-level attributes.

Label Matching: To transfer any kind of knowledge from the script corpus to the attributes from the video annotation, we need to match attribute labels to language descriptions. The annotated attribute labels are standard English verbs (for activities, e.g. "wash") and nouns (for participating objects, e.g. "carrot"), sometimes with additional particles (e.g. "take apart" and "take out"). Because the script instances contain unrestricted natural language sentences, they do not necessarily have any correspondence with the attribute label annotations, thus we evaluate two ways of mapping between them:

- **literal**: we look for the exact matching of the attribute label within the data.
- **WordNet**: we look for attribute labels and their synonyms. We take synonyms as members of the same *synset* according to the WordNet ontology [22] and restrict them to words with the same part of speech, i.e. we match only verbal synonyms to activity predicates and only nouns to object terms.

Statistics Computed on the Data: We compute two different association scores between attribute labels and composite activities:

Table 1. Dataset statistics

	videos	subjects	categories composites	categories attributes	ground truth time intervals	attribute instances	video duration
MPII Cooking	44	12	-	218	3824	15382	3-41 min
MPII Composites	212	22	41	218	8818	33876	1-23 min
combined	256	30	41	218	12642	49258	1-41 min

- **Freqs**: frequency distribution over all attribute labels for each composite activity.
- **tf∗idf**: tf∗idf (term frequency ∗ inverse document frequency, [31]) is a measure used in Information Retrieval to determine the relevance of a word for a document. Given a document collection $D = d_1, ..., d_n$, tf∗idf for a term (or word) w and a document d_i is computed as follows:

$$tf * idf(w, d_i) = freq(w, d_i) * log \frac{|D|}{|d_{w \in d}|} \tag{6}$$

  $d_{w \in d}$ is the set of documents containing w at least once. tf∗idf represents the distinctiveness of a term for a document: the value increases if the term occurs often in the document and rarely in other documents. In our case, one document corresponds to one composite activity, i.e. it contains all sequences collected for the same scenario.

We normalize the association scores for each composite activity over all attributes which gives the association weights used in Eq. 4 and 5.

5 Experimental Setup

This section first describes our new MPII Cooking Composite Activities dataset (MPII Composites) that is publicly available on our webpage. We then outline the experimental setup for the evaluation (Sec. 6).

5.1 MPII Cooking Composite Activities Dataset

To evaluate composite activity recognition, we record a dataset containing different cooking activities. We discard some of the composite activities in the script corpus (Sec. 4) which are either too elementary to form a composite activity (e.g. *how to secure a chopping board*), or were duplicates with slightly different titles, or because of limited availability of the ingredients (e.g. *butternut squash*). This resulted in 41 composite cooking activities for evaluation.

Our dataset recording setup is identical to [13] and, similarly, we do not tell subjects how to perform a certain cooking task. We compare MPII Cooking [13] and the newly proposed dataset MPII Composites in Table 1. Recordings are made with 1624x1224 pixel resolution, with 29.4fps, recording a person at the counter from the front. We use the same annotation protocol as [13], but additionally distinguish participants of an activity (*cut*), namely ingredients (*carrot*), tools (*knife*), and containers (*cutting board*), for both datasets.

Table 2. Attribute recognition using context and co-occurrence, AP in %

Attribute Training on:	MPII Cooking + MPII Composites	MPII Cooking
Base (f^0)	32.3	18.4
Context only (f^{con})	13.1	10.1
Base+Context	34.2	13.3
Co-occurrence only (f^{coocc})	27.3	20.3
Base+Co-occurrence	30.9	21.5
Base+Context+Co-occurrence	37.7	17.3

5.2 Video Representation and Evaluation Protocol

We use a bag-of-features representations which uses HOG, HOF, and motion boundary histograms around densely sampled points, which are tracked for 15 frames by median filtering in a dense optical flow field [1]. This feature showed best performance on MPII Cooking [13]. The feature extraction and training is identical to [13], i.e. we generate a codebook using k-means and train the attribute classifiers using one-vs-all SVMs trained by meanSGD [26] with a χ^2 kernel approximation [32]. We generate the codebook only from MPII Cooking, generating a true zero-shot setting when transferring to MPII Composites.

Recordings from subjects which appear in MPII Cooking are only used for training. The data of all remaining 17 subjects are divided into 6 cross-validation-splits. We report mean average precision (AP), taking the mean over all classes and cross validation rounds. If a class is not present in a cross-validation round, we exclude it from mean computation for this round.

In all evaluation runs for both attributes and composites, we use the same cross-validation procedure and we always evaluate on MPII Composites. Concerning training, we distinguish two settings: First we train attributes on both datasets (left columns, Table 2 and 3). To see how well attributes can be transferred, we also train attribute classifiers only on MPII Cooking (right columns). In the SVM case, composites are trained using meanSGD on the attribute classifier output score vector $f^{seq}(d)$.

6 Evaluation

In this section we first evaluate our attributes enhanced with context and co-occurrence, and then evaluate recognition of composite cooking activities using different levels of supervision, including a zero-shot approach using script data.

6.1 Attribute Recognition Using Context and Co-occurrence

Table 2 summarizes the results for recognizing activities and their participants, modeled as attributes. For a certain time window, multiple attributes can be activated, e.g. because a person is *mixing* a *salad* with *fork* and *spoon* in a *bowl*, resulting in 5 attributes activated at the same time.

The left column of Table 2 shows the results for training on both, MPII Cooking and MPII Composites, but evaluating on MPII Composites only. The performance of the base classifier trained on the dense trajectory feature representation achieves 32.3% mean average precision (AP) for the 218 attribute classifiers on MPII Composites.

Using only temporal context to recognize activity attributes performance drops significantly (13.1% AP). This is the expected result, because the context is similar for all activities of the same sequence and thus cannot discriminate attributes. In contrast, when using co-occurrence only, the performance drops only by 5.0% compared to the base classifiers due to the high relatedness between the attributes, namely between activities and their participants.

Combining context and co-occurrence information with the base classifier gives 34.2% and 30.9%, respectively. This is below the base classifier's performance for co-occurrence, but a combination of all training modes achieves a performance of 37.7% AP, improving the base classifier's result by 5.4%.

In a second setting, we restrict the training dataset to MPII Cooking but still evaluate on MPII Composites (right column of Table 2), requiring the activity attributes to transfer to different composite activities. When comparing the right to the left column, we notice a significant performance drop for all classifiers. This decrease can mainly be attributed to the strong reduction of training data to about one third. Co-occurrence and Base+Co-occurrence achieve the best results with 20.3% and 21.5% accuracy. Co-occurrence stand out compared to the other individual attribute classifiers: Because the activity context changes from MPII Cooking to MPII Composites (having different composite activities), context leads to tremendous performance drops in all combinations.

6.2 Composite Cooking Activity Classification

After evaluating attribute recognition performance in Sec. 6.1, we now show the results for recognizing composites using the attributes as described in Sec. 3.2. We only use the combination of base, context, and co-occurrence. Although this is not the best choice for recognizing attributes for the attribute transfer setting we found it to work better or similar to alternatives for composite recognition.

The results are shown in Table 3, which, similar to Table 2, shows results for training the attributes on both, MPII Cooking and MPII Composites, on the left and reduced attribute training on MPII Cooking only on the right. In the first (top) section of the table we use MPII Composites as training data for the composite cooking activities with 6-fold cross-validation as done before. For training of composite activities, we are limited to MPII Composites, because MPII Cooking is not structured into different composite cooking activities. In the second (bottom) section of the table we use *n*o training data for the composite cooking activities, often referred to zero-shot learning. This is enabled by the use of script data as motivated before.

The results in the top left quarter of Table 3 show the fully supervised setup. The first setup uses an SVM trained directly on the video feature representation rather than basic level attributes. This is the same setup as in [13] and as

Table 3. Composite cooking activity classification, AP in %. Top left quater: fully supervised, right column: reduced attribute training data, bottom section: no composite cooking activity training data, right bottom: true zero shot.

Attribute Training on:	MPII Cooking + MPII Composites	MPII Cooking
Training composite cooking activities on MPII Composites		
SVM (on features) [13]	38.4	-
SVM (on attributes)	51.2	32.2
NN (on attributes)	51.7	34.6
NN+Script data		
- freqs-literal	50.9	**36.2**
- freqs-WN	51.2	35.6
- tf*idf-literal	51.5	32.1
- tf*idf-WN	**53.9**	30.7
No training data for composite cooking activities		
Script data		
- freqs-literal	42.6	**22.9**
- freqs-WN	38.0	22.1
- tf*idf-literal	**49.3**	22.4
- tf*idf-WN	48.7	21.5

our Base (f^0) classifier, but this time trained and tested on complete composite activity videos. It achieves 38.4% AP, showing how challenging the dataset is. However, an SVM, trained on the attribute feature vectors (f_i^{seq}), achieves 51.2% AP, while NN classification reaches slightly better performance of 51.7%. This demonstrates that our attribute representation is a good way model for the video. To restrict NN to relevant attributes, we reduce the feature vector using script data (see Sec. 3.3). We distinguish four options: The first two use normalized frequency counts, while the third and fourth use tf*idf to determine the relevance of an attribute for a given composite. For both we mine words in the collected scripts either literally or using a WordNet (WN) expansion (see Sec. 4 for details). We first notice that tf*idf for WN (53.9%) outperforms the purely training data based methods SVM and NN. tf*idf obviously selects the right attributes for a given composite activity, making the problem of finding the nearest neighbor simpler. In comparison to the frequency counts (50.9% and 51.2%), tf*idf performs slightly better, because tf*idf activates only the most distinctive attributes for a specific composite cooking activity, while frequency counts activate less selectively based on co-occurrence of task and attribute. Comparing WordNet expansion vs. literal, we find that the expansion helps (0.3% and 2.4% increase) as it activates a broader attribute inventory.

Next we compare these results to the reduced attribute training set, leading to disjoint training set for attributes and composite cooking activities (Tab. 3, upper right quarter). Similar to the previously observed drop of performance of 20.4% for the combined attribute representation (Tab. 2, last row), we also see a significant drop in composite recognition of 19.0% and 17.1%, for SVM and NN, and 14.7% to 23.2% for the different NN+Script data versions. While the best performing approach is again based on NN+Script data, this time literal frequencies perform best with 36.2% AP. Presumably the attribute classifiers are

all too weak and select only the semantically most relevant attributes like tf*idf, but this strategy fails if these few happen to be very noisy.

In the third part (Table 3, bottom left quarter), we evaluate the case when we do not have any training labels for the composite cooking tasks which does not allow using SVM or NN. We rely on script data for selecting relevant attributes instead. Using weighted attributes (Sec. 3.3) with the same measures, we again find tf*idf to perform best with 49.3% AP for the literal version, which is a drop by only 4.6% compared to the best fully supervised case. When using frequency statistics instead of tf*idf, performance drops to 42.6% and 38.0% AP.

Finally, we show our results on a true zero-shot setting (Table 3, right bottom part). We would like to stress that the attributes have only been trained on MPII Cooking and not as part of the unseen composites, nor are feature representations or composite cooking activities trained for the new MPII Composites, and also subjects are disjoint. Associations to unknown data is only provided by script data and not manually defined. For this challenging setting, we achieve a performance of 22.9% AP for the freqs-literal measure outperforming again the others like for the supervised case above.

Overall we found that script data improves performance by 2.2% AP to 53.9% AP in the fully supervised case and by 1.6% to 36.2% AP for reduced attribute training data. It also enables recognizing highly varied cooking tasks without training data close to supervised performance (49.3%) and obtains encouraging 22.9% for the complete zero-shot case where training happens entirely on a different dataset, different people, and different cooking tasks.

7 Conclusion

Composite activities are difficult to recognize because of their inherent variability and the lack of training data for specific composites. This paper shows that attribute-based activity recognition allows recognizing composite activities well. Most notably, we have shown how textual script data, which is easy to collect, enables an improvement of the composite activity recognition when only little training data is available, and even allows for complete zero-shot transfer. We have also shown that activity attribute recognition can be improved by using context and co-occurrence attributes. A direction for future work is to use the mined textual descriptions to produce detailed textual descriptions of a video sequence, and exploit more of the script structure for action recognition.

References

1. Wang, H., Kläser, A., Schmid, C., Liu, C.-L.: Action Recognition by Dense Trajectories. In: CVPR (2011)
2. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: CVPR (2010)
3. Niebles, J.C., Chen, C.-W., Fei-Fei, L.: Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 392–405. Springer, Heidelberg (2010)
4. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR (2004)

5. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
6. Liu, J.G., Luo, J.B., Shah, M.: Recognizing realistic actions from videos 'in the wild'. In: CVPR (2009)
7. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR (2008)
8. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV (2009)
9. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities, cvpr. In: ICCV (2011)
10. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
11. Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I., Schiele, B.: What Helps Where – And Why? Semantic Relatedness for Knowledge Transfer. In: CVPR (2010)
12. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR (2011)
13. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: CVPR (2012)
14. Laptev, I.: On space-time interest points. In: IJCV (2005)
15. Chakraborty, B., Holte, M.B., Moeslund, T.B., Gonzalez, J., Roca, F.X.: A selective spatio-temporal interest point detector for human action recognition in complex scenes. In: ICCV (2011)
16. Gupta, A., Davis, L.S.: Objects in action: An approach for combining action understanding and object perception. In: CVPR (2007)
17. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.M.: A scalable approach to activity recognition based on object use. In: ICCV (2007)
18. Li, L.J., Su, H., Lim, Y., Fei-Fei, L.: Objects as attributes for scene classification. In: ECCV (2010)
19. Yao, B., Jiang, X., Khosla, A., Lin, A.L., Guibas, L., Fei-Fei1, L.: Human action recognition by learning bases of action attributes and parts. In: ICCV (2011)
20. Ferrari, V., Zisserman, A.: Learning visual attributes. In: NIPS (2007)
21. Farhadi, A., Endres, I., Hoiem, D.: Attribute-centric recognition for cross-category generalization. In: CVPR (2010)
22. Fellbaum, C.: WordNet: An Electronical Lexical Database. The MIT Press (1998)
23. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
24. Snoek, C., Worring, M., van Gemert, J., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: ACM Multimedia (2006)
25. Hauptmann, A.G., Christel, M.G., Yan, R.: Video retrieval based on semantic concepts. Proceedings of IEEE 96 (2008)
26. Rohrbach, M., Stark, M., Schiele, B.: Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In: CVPR (2011)
27. Schank, R.C., Abelson, R.P.: Scripts, Plans, Goals and Understanding (1977)
28. Barr, A., Feigenbaum, E.: The Handbook of Artificial Intelligence, vol. 1. William Kaufman Inc., Los Altos (1981)
29. Regneri, M., Koller, A., Pinkal, M.: Learning script knowledge with web experiments. In: Proceedings of ACL 2010 (2010)
30. Bloem, J., Regneri, M., Thater, S.: Robust processing of noisy web-collected data. In: Proceedings of KONVENS 2012 (2012)
31. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. In: Information Processing and Management (1988)
32. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. In: CVPR (2010)