Action Sets: Weakly Supervised Action Segmentation without Ordering Constraints

Alexander Richard, Hilde Kuehne, Juergen Gall
University of Bonn, Germany
{richard,kuehne,gall}@iai.uni-bonn.de

Abstract

Action detection and temporal segmentation of actions in videos are topics of increasing interest. While fully supervised systems have gained much attention lately, full annotation of each action within the video is costly and impractical for large amounts of video data. Thus, weakly supervised action detection and temporal segmentation methods are of great importance. While most works in this area assume an ordered sequence of occurring actions to be given, our approach only uses a set of actions. Such action sets provide much less supervision since neither action ordering nor the number of action occurrences are known. In exchange, they can be easily obtained, for instance, from meta-tags, while ordered sequences still require human annotation. We introduce a system that automatically learns to temporally segment and label actions in a video, where the only supervision that is used are action sets. An evaluation on three datasets shows that our method still achieves good results although the amount of supervision is significantly smaller than for other related methods.
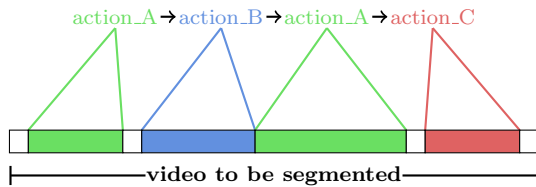
1. Introduction

Due to the huge amount of publicly available video data, there is an increasing interest in methods to analyze these data. In the field of human action recognition, considerable advances have been made in recent years. A lot of research has been published on action recognition, *i.e.* action classification on pre-segmented video clips [35, 30, 11]. While current methods already achieve high accuracies on large datasets such as UCF-101 [32] and HMDB-51 [14], the assumption of having pre-segmented action clips does not apply for most realistic tasks. Therefore, there is a growing interest in efficient methods for finding actions in temporally untrimmed videos. With the availability of large scale datasets such as Thumos [9], Activity Net [5], or Breakfast [12], many new approaches to temporally locate and classify actions in untrimmed videos

emerged [27, 21, 39, 29, 24, 18]. However, these approaches usually rely on fully supervised data, *i.e.* the exact temporal location of each action occurring in the training videos is known. Creation of such training data requires manual annotation on video frame level which is very expensive as well as impractical for large datasets. Thus, there is a need for methods that can learn temporal action segmentation and labeling with less supervision. A commonly made assumption is that instead of full supervision, only an ordered sequence of the actions occurring in the video is provided [2, 15, 6, 25]. Although this kind of weak supervision is already much easier to obtain, *e.g.* from movie scripts or subtitles, for a vast amount of real world tasks, such information still can not be assumed to be available. Instead, weak labels often arise in form of meta tags or unordered lists from document indexing.

To address this problem, we propose a weakly supervised method that can learn temporal action segmentation and labeling from unordered action labels, which we refer to as action sets. In contrast to the above mentioned methods (*cf.* Figure 1a), we assume that neither ordering nor number of occurrences of actions is provided during training. Instead, only a set of actions occurring within the video is given (*cf.* Figure 1b). This task is much more difficult than the case where ordered action transcripts are given. Consider, for instance, a video with T frames and a transcript of C ordered actions. Then, there are $\frac{(C+T)!}{C!T!}$ possible labelings for the video. If the actions are not ordered, there are already C^T possible labelings. For a very short video of 100 frames and $C = 5$, this means that using unordered actions sets as supervision already allows for about 10^{60} times more possible labelings than when provided ordered action transcripts.

In order to deal with such an enormously large search space, we propose three model components that aim at decomposing the search space on three different levels of granularity. The coarsest level is addressed by a context model that restricts the space of possible action sequences. On a finer level, a length model restricts the durations of actions to a reasonable length. On the lowest, most fine-

(a) weak supervision: ordered action sequences

action_A → action_B → action_A → action_C

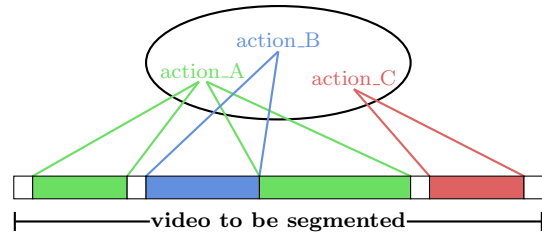**(b) weak supervision: action sets**

action_A action_B action_C

Figure 1. (a) Weak supervision with ordered action sequences [2, 15, 6]. The number of actions and their ordering is known. (b) Weak supervision with action sets (our setup). Note that neither action orderings nor the number of occurrences per action are provided.

grained level, a frame model provides class probabilities for each video frame.

Note that context models [15, 25] and length models [24] have been used before. However, in these works either ordered action transcripts for the context model or framewise annotations for the length models are provided. To the best of our knowledge, we are the first to use these models without being provided any training data that allows to directly infer such models from the video annotation.

In an extensive evaluation, we investigate the impact of each component within the system. Moreover, temporal segmentation and action labeling quality is evaluated on unseen videos alone and on videos with action sets given at inference time as additional supervision.

2. Related Work

Strong feature extractors developed in classical action recognition such as Fisher vectors of improved dense trajectories [35] or a variety of sophisticated CNN methods [30, 8, 11, 4] have also pushed the advances in untrimmed action segmentation.

When processing untrimmed videos, actions can either be localized in the temporal domain only [2, 24, 39, 29, 25, 15, 17, 26], or in the spatio-temporal domain [33, 7, 20, 38]. For the latter, videos are usually constrained to contain only few action instances. While most approaches in this area are fully supervised, [38] propose a weakly supervised method for actor-action segmentation that is based on a multi-task ranking model.

In this work, we focus on localizing actions in the temporal domain only. In this setting, videos either contain multiple actions of several classes occurring densely throughout the whole video [12, 27], or sparsely [9], *i.e.* most of the video is background and all instances of a single class or a small set of classes need to be detected in the video. Well studied methods from classical action recognition are frequently used as framewise feature extractors [21, 27, 39, 24]. Although CNN features are successful in some action detection methods [39, 29], they usually require to be retrained using full supervision. Improved dense trajectories, on the contrary, are extracted in an unsuper-

vised manner, making them the features of choice for most weakly supervised approaches [2, 15, 6, 25].

In the context of fully supervised action detection, most approaches use a sliding window to efficiently segment a video [21, 27] and rely on CNNs or recurrent networks [39, 31, 29] that can not be used if only weak supervision is available. The same holds for [24], who model context and length information, which is also done in our approach. They show that length and context information significantly improve action segmentation systems, using a Poisson distribution to model action lengths and a language model to incorporate action context information. Other fully supervised methods guided by grammars have been proposed in [22, 34, 13]. Note that in contrast to our task, their length and context model can be easily estimated from the frame-level training annotations. The challenge for our problem formulation, however, is that no annotations that allow a direct estimation of a context or length model are provided.

When working with weak supervision, existing methods use ordered action sequences as annotation. Early works suggest to get action sequences from movie scripts [16, 3]. Alayrac *et al.* [1] propose to localize specific actions in a video from narrated instructions. In [19], it is proposed to use automatic speech recognition and align textual descriptions, in their cases recipes, to the recognized spoken sequence. Bojanowski *et al.* [2] address the task of aligning actions to frames. In their work, ordered action sequences are assumed to be provided during training and testing and only an alignment between the frames and the action sequence is learned. Kuehne *et al.* [15] extend their approach from [13] to weak supervision by inferring a linear segmentation from ordered action sequences and training a classical GMM+HMM speech recognition system on iteratively refined segmentations. A further extension of this idea has been proposed by Richard *et al.* [25], where the GMM is replaced by a recurrent neural network. Recently, Huang *et al.* [6] proposed to use connectionist temporal classification (CTC) to learn temporal action segmentation from weakly supervised videos. In order to avoid degenerate alignments between video frames and provided action transcripts, they

propose to use a visual similarity measure as an extension to the classical CTC approach.

In contrast to the approaches of [15, 2, 6, 25], our approach only uses action sets, *i.e.* a much weaker supervision. Consequently, the way our model is learned is also different from the above mentioned approaches.

Another recently published and related method by Wang *et al.* [36] addresses the task of detecting an action in a video with sparse action occurrences. More precisely, for a given action class, they generate action proposals and train a neural network to distinguish instances of this action from background in the video. Being designed to distinguish actions from background in a video, their method is not suited for densely labeled videos containing many difference actions followed by one another, as it is the case in this paper.

3. Temporal Action Labeling

Task Definition. Let $(x_1, \ldots, x_T)$ be a video with T frames and x_t are the framewise feature vectors. The task is to assign an action label c from a predefined set of possible labels $\mathcal{C}$ to each frame of the video. Following the notation of [24], connected frames of the same label can be interpreted as an action segment of class c and length l. With this notation, the goal is to cut the video into an unknown number of N action segments, *i.e.* to define N segments with lengths $(l_1, \ldots, l_N)$ and action labels $(c_1, \ldots, c_N)$. To simplify notation, we abbreviate sequences of video frames, lengths, and classes by $\mathbf{x}_1^T$, $\mathbf{l}_1^N$, and $\mathbf{c}_1^N$, where the subscript is the start index of the sequence and the superscript the ending index.

Model Definition. In order to solve this task, we propose a probabilistic model and aim to find the most likely segmentation and segment labeling of a given video,

$$(\hat{\mathbf{l}}_1^N, \hat{\mathbf{c}}_1^N) = \arg\max_{N, \mathbf{l}_1^N, \mathbf{c}_1^N} \left\{ p(\mathbf{c}_1^N, \mathbf{l}_1^N | \mathbf{x}_1^T) \right\}, \qquad (1)$$

where l_n is the length of the n-th segment and c_n is the corresponding action label. We use a background class for all parts of the video in which no action (or no action of interest) occurs. So, all video frames belong to one particular action class and segment. Hence, $\mathbf{l}_1^N$ and $\mathbf{c}_1^N$ define a segmentation and labeling of the complete video.

In order to build a probabilistic model, we first decompose Equation (1) using Bayes rule,

$$(\hat{\mathbf{l}}_1^N, \hat{\mathbf{c}}_1^N) = \arg\max_{N, \mathbf{l}_1^N, \mathbf{c}_1^N} \left\{ p(\mathbf{c}_1^N) p(\mathbf{l}_1^N | \mathbf{c}_1^N) p(\mathbf{x}_1^T | \mathbf{c}_1^N, \mathbf{l}_1^N) \right\}. \qquad (2)$$

The first factor, $p(\mathbf{c}_1^N)$ is the coarsest model, controlling the likelihood of action sequences. The second factor, on a finer level, is a length model that controls the action durations, and the third factor finally provides a likelihood of the

video frames for a specific segmentation and labeling. The same factorization has also been proposed in [24] for fully supervised action detection. We would like to emphasize that our model only shares the factorization with the work of [24]. Due to weak supervision, the actual models we use and the way they are trained are highly different.

3.1. Weak Supervision

While most works on weakly supervised temporal action segmentation use ordered action sequences as supervision [6, 15, 2], in our task, only unordered sets of actions occurring in the video are provided, *cf*. Figure 1b. Notably, neither the order of the actions nor the number of occurrences per action is known. Assuming the training set consists of I videos, then the supervision available for the i-th video is a set $\mathcal{A}_i \subseteq \mathcal{C}$ of actions occurring in the video.

During inference, no action sets are provided for the video and the model has to infer an action labeling from the video frames only. As an additional task, we also discuss the case where action sets are given for inference, see Section 4.6.

In the following, the models for the three factors $p(\mathbf{c}_1^N)$, $p(\mathbf{l}_1^N | \mathbf{c}_1^N)$, and $p(\mathbf{x}_1^T | \mathbf{c}_1^N, \mathbf{l}_1^N)$ from Equation (2) are introduced.

3.2. Context Modeling with Context-free Grammars

Our first step to handle the huge search space is to restrict the possible action orderings using a context-free grammar $\mathcal{G}$ in order to model the context prior $p(\mathbf{c}_1^N)$. Once the grammar is generated, define

$$p(\mathbf{c}_1^N) = \begin{cases} \text{const,} & \text{if } \mathbf{c}_1^N \in \mathcal{G}, \\ 0, & \text{otherwise.} \end{cases} \qquad (3)$$

Concerning the maximization in Equation (2), this means that each action sequence generated by $\mathcal{G}$ has the same probability and all other sequences have zero probability, *i.e.* they can not be inferred. We propose the following strategies to obtain a grammar:

Naive Grammar. All action sequences that can be created using elements from each action set from the training data are possible. Formally, this means

$$\mathcal{G}_{\text{naive}} = \bigcup_{i=1}^{I} \mathcal{A}_i^*, \qquad (4)$$

where i indicates the i-th training sample and $\mathcal{A}_i^*$ is the Kleene closure of $\mathcal{A}_i$.

Monte-Carlo Grammar. We randomly generate a large amount of k action sequences. Each sequence is generated by randomly choosing a training sample $i \in \{1, \ldots, I\}$. Then, actions are uniformly drawn from the corresponding

action set $\mathcal{A}_i$ until the accumulated estimated means λ_c of all drawn actions exceed the video length T_i. The mean lengths λ_c are estimated action class durations, see Section 3.3.

Text-Based Grammar. Frequently, it is possible to obtain a grammar from external text sources, *e.g.* from web recipes or books. Given some natural language texts, we enhance the monte-carlo grammar by mining frequent word combinations related to the action classes. Consider two action classes v and w, for instance `butter_pan` and `crack_egg`. If either of the words `butter` or `pan` is preceding `crack` or `egg` in the textual source, we increase the count $N(v, w)$ by one. This way, word conditional probabilities

$$p(w|v) = \frac{N(v, w)}{\sum_{\tilde{w}} N(v, \tilde{w})} \qquad (5)$$

are obtained that have a high value if v precedes w frequently and a low value otherwise. The actual construction of the grammar follows the same protocol as the monte-carlo grammar with the only difference that the actions are not drawn uniformly from the action set but according to the distribution $p(w|v)$, where v is the previously drawn action class.

3.3. Length Model from Action Sets

While a grammar already introduces some ordering constraints, the search space is still tremendously large, considering that actions can be of arbitrary and even practically unreasonable durations. Therefore, as a second step, we estimate a length model out of the scarce information we get from the training data. In order to model the length factor $p(\mathbf{l}_1^N | \mathbf{c}_1^N)$, we assume conditional independence of each segment length and further drop the dependence of all class labels but the one of the current segment, *i.e.*

$$p(\mathbf{l}_1^N | \mathbf{c}_1^N) = \prod_{n=1}^{N} p(l_n | c_n). \qquad (6)$$

Each class-conditional $p(l|c)$ is modeled with a Poisson distribution for class c.

For the estimation of the class-wise Poisson distributions, only the action sets $\mathcal{A}_i$ provided in the training data can be used. Ideally, the free parameter of a Poisson distribution, λ_c, should be set to the mean length of action class c. Since this can not be estimated from the action sets, we propose two strategies to approximate the mean duration of each action class.

Naive Approach. In the naive approach, the frames of each training video are assumed to be uniformly distributed among the actions in the respective action set. The average length per class can then be computed as

$$\lambda_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \frac{T_i}{|\mathcal{A}_i|}, \qquad (7)$$

where $\mathcal{I}_c = \{i : c \in \mathcal{A}_i\}$ and T_i is the length of the i-th video.

Loss-based. The drawback of the naive approach is that actions that are usually short are assumed to be longer if the video is long. Instead, we propose to estimate the mean of all classes together. This can be accomplished by minimizing a quadratic loss function,

$$\sum_{i=1}^{I} \sum_{c \in \mathcal{A}_i} (\lambda_c - T_i)^2 \quad \text{subject to } \lambda_c > l_{\min}, \qquad (8)$$

where $l_{\min}$ is a minimal action length. For minimization, we use constrained optimization by linear approximation (COBYLA) [23].

Note that the true mean length of action c is likely to be smaller than λ_c since actions may occur multiple times in a video. However, this can not be included into the length model since the action sets do not provide such information.

3.4. Multi-task Learning of Action Frames

Given the grammar and the length model that already strongly restrict the search space, the last missing factor is the actual framewise model providing a likelihood for each class to be present in a given frame.

In order to model this last factor from Equation (2), we train a network with $|\mathcal{C}|$ many binary softmax output layers. Each layer predicts if for a given frame x_t label c is present, *i.e.* if $c \in \mathcal{A}_i$ or not. Since an action c usually occurs in different context, all frames belonging to class c are always labeled with its true class c and some varying other classes. Thus, a classifier can learn a strong response on the presence of the correct class and weaker responses on the presence of other falsely assigned classes. As loss of our network, we therefore use the accumulated cross-entropy loss of each binary classification task.

In order to use the output probabilities of the multi-task network during inference, they need to be transformed to model the last factor from Equation (2), $p(\mathbf{x}_1^T | \mathbf{c}_1^N, \mathbf{l}_1^N)$. We therefore define the class-posterior probabilities

$$p(c|x_t) := \frac{p(c\ present | x_t)}{\sum_{\tilde{c}} p(\tilde{c}\ present | x_t)} \qquad (9)$$

and transform them into class-conditional probabilities

$$p(x_t|c) \propto \frac{p(c|x_t)}{p(c)}. \qquad (10)$$

Since the network is a framewise model, $p(c)$ is also a framewise prior. More specifically, if $\mathrm{count}(c)$ is the total number of frames labeled with c *present*, then $p(c)$ is the relative frequency $\mathrm{count}(c)/\sum_{\tilde{c}} \mathrm{count}(\tilde{c})$.

Assuming conditional independence of the video frames, the probability of an action segment ranging from frame t_s to t_e can then be modeled as

$$p(\mathbf{x}_{t_s}^{t_e}|c) = \prod_{t=t_s}^{t_e} p(x_t|c). \qquad (11)$$

Framewise conditional independence is a commonly made assumption in multiple action detection and temporal segmentation methods [24, 15, 13]. Note that t_s and t_e are implicitly given by the segment lengths $\mathbf{l}_1^N$. For the n-th segment in the video, $t_s^{(n)} = 1 + \sum_{i<n} l_i$ and $t_e^{(n)} = \sum_{i\leq n} l_i$.

The third factor of Equation (2) is now modeled using the previously defined segment probabilities,

$$p(\mathbf{x}_1^T|\mathbf{c}_1^N, \mathbf{l}_1^N) := \prod_{n=1}^{N} p(\mathbf{x}_{t_s^{(n)}}^{t_e^{(n)}}|c_n). \qquad (12)$$

3.5. Inference

With the explicit models for each factor, the optimization problem from Equation (2) reduces to

$$(\hat{\mathbf{l}}_1^N, \hat{\mathbf{c}}_1^N) = \underset{N, \mathbf{l}_1^N, \mathbf{c}_1^N \in \mathcal{G}}{\arg\max} \Big\{ \prod_{n=1}^{N} p(l_n|c_n) \cdot p(\mathbf{x}_{t_s^{(n)}}^{t_e^{(n)}}|c_n) \Big\}. \qquad (13)$$

Note that the $\arg\max$ is only taken over action sequences that can be generated by the grammar. Since the same probability has been assigned to all those sequences, the factor $p(\mathbf{c}_1^N)$ from Equation (2) is a constant. Moreover, the length model $p(l_n|c_n)$ strongly penalizes unlikely action durations and allows for an efficient pruning of unlikely segmentations. Both together lead to a significant reduction of the search space.

The solution to Equation (13) can now be efficiently computed using a Viterbi algorithm over context-free grammars, as widely used in automatic speech recognition, see for example [10]. The algorithm is linear in the number of frames and therefore allows for efficient processing of videos with arbitrary length. The authors of [24] have shown that adding a length model increases the complexity from $\mathcal{O}(T)$ to $\mathcal{O}(TL)$, where L is the maximal action length that can occur. In theory, there is no limitation on the duration of actions, so inference would be quadratic in the number of frames. In practice, however, it is usually possible to limit the maximal allowed action length L to some reasonable constant, maintaining linear runtime.[1]

4. Experiments

In this section, we analyze the components of our approach, starting with the grammar (Section 4.2) and the

---

[1]Source code and details on the dynamic programming equations can be found on https://alexanderrichard.github.io

length model (Section 4.3), before we compare our system to existing methods that use more supervision (Section 4.5).

4.1. Setup

Datasets. We evaluate our approach on three datasets for weakly supervised temporal action segmentation and labeling, namely the Breakfast dataset [12], MPII Cooking 2 [28], and Hollywood Extended [2].

The **Breakfast** dataset is a large scale dataset comprising $1,712$ videos, corresponding to roughly 67 hours of video and 3.6 million frames. Each video is labeled by one of the 10 coarse breakfast related activities like *coffee* or *fried eggs*. Additionally, a finer action segmentation into 48 classes is provided which is usually used for action detection and segmentation. Overall, there are nearly $12,000$ instances of these fine grained action classes with durations between a few seconds and several minutes, making the dataset very challenging. The actions are densely annotated and only 7% of the frames are background frames. We use four splits as suggested in [12] and provide frame accuracy as evaluation metric.

MPII Cooking 2 consists of 273 videos with 2.8 million frames. We use the 67 action classes without object annotations. Overall, around $14,000$ action segments are annotated in the dataset. The dataset provides a fixed split into a train and test set, separating 220 videos for training. With 29%, the background portion in this dataset is at a medium level. For evaluation, we use the midpoint hit criterion as proposed in [27].

Hollywood Extended is a smaller dataset comprising 937 videos with roughly $800,000$ frames. There are about $2,400$ non-background action instances from 16 different classes. With 61% of the frames, the background portion within this dataset is comparably large. We follow the suggestion of [2] and use a 10-fold cross-validation. The originally proposed evaluation metric is a variant of the Jaccard index, intersection over detection, which is only reasonable for a transcript-to-video alignment task where the transcripts and thus the action orderings are known for the test sequences as in [2] and [6]. For temporal action segmentation, only a video is given during inference and the number of predicted segments can differ from the number of annotated segments. In this case, the metric can not be used. Thus, we stick to the Jaccard index (intersection over union), which is widely used in the domain of action detection [24, 9] and has also been used on this dataset by [15].

Feature extraction. For a fair comparison, we use the same features as [15] and [6]. Fisher vectors of improved dense trajectories [35] are extracted for each frame and the result is projected to a 64-dimensional subspace using PCA as proposed by Kuehne *et al.* [13]. Then, the features are normalized to have zero mean and unit variance along each dimension. If not mentioned otherwise, we use the monte-

Grammar	frame accuracy	
	train	test
none	0.147	0.099
naive	0.194	0.134
monte-carlo	0.282	0.233
manually created	0.333	0.269
ground truth	0.367	0.294

Table 1. Evaluation of our method on Breakfast using different context-free grammars. As length model, the loss-based approach is used.

carlo grammar and the loss-based length model. The in-depth evaluation of our approach is conducted on Breakfast, final results on other datasets are reported in Section 4.5.

Model. For the neural network in the framewise model we use a simple feed forward network with a single hidden layer of 256 rectified linear units. Experiments with deeper models could not generalize to the test data (VGG-16: accuracy of 0.031 on Breakfast). We also evaluated the neural network based multiple instance learning approach of [37], which also was not able to make reliable predictions (accuracy 0.089 on Breakfast). We therefore found the multi-task network as proposed in Section 3.4 to be a simple yet effective model.

Efficient inference. During inference, we allow to hypothesize new segments only every 30 frames. This allows for inference roughly in realtime without affecting the performance of the system compared to a more fine-grained segment hypothesis generation.

4.2. Effect of the Grammar

The main contribution of the grammar is to limit the search space and remove unrealistic action sequences. We compare different kinds of grammars and report the frame accuracy on both, test and train set. Recall that due to weak supervision, our method does not necessarily provide good results on the training videos, making it interesting to investigate both sets. As shown in Table 1, the use of a sophisticated grammar is crucial for good performance. The naive grammar is only slightly better than the system without any grammar. The monte-carlo grammar boosts the frame accuracy by 10% on the test set. Note that we found the number of k monte-carlo samples for the grammar not to be critical and chose $1,000$ randomly generated sequences for all experiments. Using a ground truth grammar, *i.e.* a grammar learned from ordered action transcripts (which are not provided in our setting) gives an upper bound on the performance that can be reached by improving the grammar only. Notably, the monte-carlo grammar is only 6% below this upper bound.

For a further comparison, we gave all action sets from the training data to an annotator who was asked to manually create an ordered action sequence for each set. This manually created grammar serves as a comparison of the purely

	Breakfast *frame acc.*	Cooking 2 *midpoint hit*	Holl. Ext. *jacc. idx*
monte-carlo	0.233	0.098	0.093
text-based	0.232	0.106	0.092

Table 2. Evaluation of the text-based grammar. For Cooking 2, where the text sources are closely related to the content of the videos, an improvement can be observed.

Length model	frame accuracy	
	train	test
naive	0.254	0.201
loss-based	0.282	0.233
ground truth	0.341	0.257

Table 3. Evaluation of our method on Breakfast using different length models. As grammar, the monte-carlo approach is used.

data driven monte-carlo grammar to human knowledge. Although the manual grammar is better, the frame accuracy only differs by 3.6%. Since the annotator on average only needed one minute per action set, a manual grammar is also a cheap opportunity to add human knowledge without the need to actually annotate videos.

As proposed in Section 3.2, textual sources can be used to enhance the monte-carlo grammar by restricting the transition between action classes to only the likely ones. We evaluate such a text-based grammar for all three datasets. For Breakfast, we used a webcrawler to download more than $1,200$ breakfast related recipes, for Hollywood Extended, 10 movie scripts of IMDB top-ranked movies have been downloaded, and for Cooking 2, we used the scripts provided by the authors of the dataset. These scripts were obtained by asking annotators to write sequential instructions on how to execute the respective kitchen task. Consequently, the text sources used for Breakfast and Hollywood Extended are only loosely connected to the datasets, whereas the textual source for Cooking 2 covers exactly the same domain as the videos. Not surprisingly, we find that only for this case, the text-based grammar leads to an improvement over the monte-carlo grammar, *cf*. Table 2. For the other datasets, neither an improvement nor a degradation is observed.

4.3. Effect of the Length Model

Besides the choice of the context-free grammar, the length model is a crucial component of our system. The estimated mean action lengths influence the performance in two ways: first, they define the Poisson distribution that contributes to the actual length of hypothesized action segments. Secondly, they have a huge impact on the number of action instances that are generated for each action sequence in the monte-carlo grammar.

Mean Length Approximation. We compare the two proposed mean approximation strategies, naive and loss-based mean approximation, with a ground truth model, *i.e.*

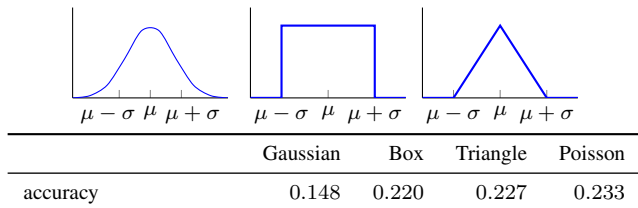|	Gaussian	Box	Triangle	Poisson
accuracy	0.148	0.220	0.227	0.233

Figure 2. Evaluation of different length models on Breakfast.

the true action means estimated on a frame-level ground truth annotation of the training data. The results are shown in Table 3. The naive mean approximation suffers from some conceptual drawbacks. Due to the uniform distribution of video frames among all actions occurring in the video, short actions may be assigned a reasonable length as long as the video is also short. If the video is long, however, short actions get the same share of frames as long actions, resulting in an over-estimation of the mean for short actions and an under-estimation of the mean for long actions. The loss-based mean approximation, on the contrary, can provide more realistic estimates by minimizing Equation (8). Note that the solution of the problem in principle would allow for negative action means. Hence, setting the minimal action length $l_{\min} > 0$ is crucial. In practice, we want to ensure a reasonable minimum length and set $l_{\min} = 50$ frames, corresponding to roughly two seconds of video. The loss-based mean approximation performs significantly better than the naive approximation, increasing the frame accuracy by 3%.

Comparing these numbers to the ground truth length model reveals that particularly on the train set, on which the ground truth lengths have been estimated, there is still room for improvement. Considering the small amount of supervision that we can utilize to estimate mean lengths, *i.e.* actions sets only, and the small gap between the loss-based approach and the ground truth model on the test set, on the other hand, we find that our loss-based method already yields a good approximation.

Evaluating Different Length Models. So far we modeled the length with a Poisson distribution. There is a variety of other possible length models. In Figure 2, three additional models are evaluated, a Gaussian, a box-, and a triangle model. Box and triangle model are zero outside $[\mu - \sigma, \mu + \sigma]$. The standard deviation σ of each action class is heuristically estimated by mapping actions according to their mean length onto the possible segmentations generated by the monte-carlo grammar. The Gaussian model decays too fast around the mean lengths and leads to low accuracies. Although the other models perform well, the Poisson distribution still yields the best results.

4.4. Impact of Model Components

All three components, the grammar, the length model, and the framewise model, contribute their share to restrict-

		frame accuracy	
grammar	length model	train	test
✗	✗	0.118	0.080
✗	✓	0.147	0.099
✓	✗	0.208	0.154
✓	✓	0.282	0.233
fully supervised		0.774	0.556

Table 4. The first four rows are a comparison of the impact of the grammar and the length model on the Breakfast dataset; the last is our system trained on fully supervised, *i.e.* framewise annotated, data. It is an upper bound for the weakly supervised setup.
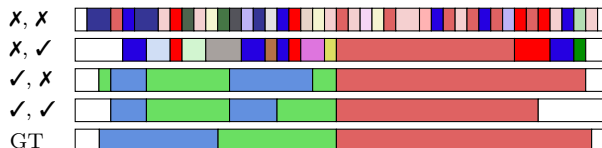

Figure 3. Example segmentation on a test video from Breakfast. Row one to four correspond to row one to four from Table 4. The last row is the ground truth segmentation.

ing the search space to reasonable segmentations. In this section, we evaluate the impact of the grammar and length model on their own and in combination with each other. We use the best-working grammar and length approximation, *i.e.* the monte-carlo grammar with loss-based mean approximation, and analyze the effect of omitting the grammar and/or the length model from Equation (13) during inference. The results are reported in Table 4. Not surprisingly, the performance without a grammar is poor, as the model easily hypothesizes unreasonable action sequences. Adding a grammar alone already boosts the performance, restricting the search space to more reasonable sequences. In order to also get action segments of reasonable length, however, the combination of grammar and length model is crucial. This effect can also be observed in a qualitative segmentation result, see Figure 3. Note the strong over-segmentation if neither grammar nor length model is used. Introducing the length model partially improves the result but still the grammar is crucial for a reasonable segmentation in terms of correct segment labeling and segment lengths. The fully supervised model (last row of Table 4) is trained by assigning the ground truth action label to each video frame. Apart from the labeling, the multi-task network architecture remains unchanged. The full supervision defines an upper bound for our weakly supervised method.

4.5. Comparison to State of the Art

The task of weakly supervised learning of a model for temporal action segmentation given only action sets has not been addressed before. Still, there are some works on temporal action segmentation given ordered action sequences. In this section, we compare our approach to these methods on the three datasets. Kuehne *et al.* [15] approach the problem with hidden Markov models and Gaussian mixture

	Breakfast *frame acc.*	Cooking 2 *midpoint hit*	Holl. Ext. *jacc. idx*
Weak supervision: unordered action sets			
monte-carlo	0.233	0.098	0.093
text-based	0.232	0.106	0.092
Stronger supervision: ordered action transcripts			
HMM [15]	0.259	0.200	0.086
CTC [6]	0.218	–	–
ECTC [6]	0.277	–	–
HMM+RNN [25]	0.333	–	0.119

Table 5. Performance of our method compared to state of the art methods for weakly supervised temporal segmentation. Note that our method uses action sets as weak supervision, whereas [15, 6, 25] have a stronger supervision with ordered action sequences.

Figure 4. Example segmentation. All relevant ground truth actions are present. Note that *spoon_powder* always occurs jointly with *pour_milk*, so it is hard for our model to distinguish them.

cuts per video	4	2	-
avg. #actions per video	12.5	25	50
midpoint hit	0.174	0.121	0.098

Table 6. Different levels of video trimming for Cooking 2. More videos and less actions per video result in better performance.

models and Richard *et al.* [25] extend their approach using recurrent neural networks. Huang *et al.* [6], in contrast, rely on connectionist temporal classification (CTC) with LSTMs and extend it by downweighting degenerated alignments and incorporating visual similarity of frames into the decoding algorithm. They call their approach extended CTC (ECTC). All of these approaches use ordered action sequences, and thus a much stronger supervision than our method. Keeping the tremendously large search space for our problem compared to [15, 6, 25] in mind (*cf*. Section 1), our model achieves remarkable results on Breakfast and Hollywood Extended, *cf*. Table 5. Note that training the HMM approach of [15] with monte-carlo sampled action transcripts (*i.e.* with the same amount of supervision as in this paper) only yields an accuracy of 0.145 on Breakfast, which is far less than our approach. An example segmentation of our approach is shown in Figure 4. Falsely recognized actions are frequently those that only occur jointly, such as spoon_powder and pour_milk. In these cases, the model typically fails to predict the correct ordering.

Actions per Video. While our approach works well on Breakfast and Hollywood Extended, the results on Cooking 2 show its limitations. The dataset has many classes (67) but only a small amount of training videos (220), which are very long and contain a huge amount of different ac-

	Breakfast *frame acc.*	Cooking 2 *midpoint hit*	Holl. Ext. *jacc. idx*
monte-carlo	0.284	0.102	0.230
text-based	0.280	0.106	0.242

Table 7. Results of our method when the action sets are provided for inference.

tions. These characteristics make it difficult for the multi-task learning to distinguish different classes, as many of them occur jointly in most training videos. We show the importance of having enough videos by cutting each video of Cooking 2 into two/four parts (Table 6). This increases the number of videos and reduces the number of actions per video. The more videos and the less actions per video on average, the better are the results of our method.

4.6. Inference given Action Sets

So far, it has always been assumed that no weak supervision in form of action sets is provided for inference. If the action sets for the videos are, for example, generated using meta-tags of Youtube videos, however, they may as well be available during inference. In this section, we evaluate our method under this assumption.

Let $\mathcal{A}$ be the given action set for a video. During inference, only action sequences that are consistent with $\mathcal{A}$ need to be considered, *i.e.* for a grammar $\mathcal{G}$, only sequences $\mathbf{c}_1^N \in \mathcal{G} \cap \mathcal{A}^*$ are possible. If $\mathcal{G} \cap \mathcal{A}^*$ is empty, we consider all sequences $\mathbf{c}_1^N \in \mathcal{A}^*$. The results are shown in Table 7. The above mentioned limitations on Cooking 2 again prevent our method from generating a better segmentation. On Breakfast and Hollywood Extended, a clear improvement of 5% and 15% compared to the inference without given action sets (Table 5) can be observed.

5. Conclusion

We have introduced a system for weakly supervised temporal action segmentation given only unordered action sets. In contrast to ordered action sequences that have been proposed as weak supervision by previous works, action sets are often publicly available in form of meta-tags for videos and do not need to be annotated. Although action sets provide by far less supervision than ordered action sequences and lead to a tremendously large search space, our method still achieves good results. Providing the possibility to incorporate data-driven grammars as well as text-based information or human knowledge, our method can be adapted to specific requirements in different video analysis tasks.

References

[1] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016. 2

[2] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conf. on Computer Vision*, pages 628–643, 2014. 1, 2, 3, 5

[3] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Int. Conf. on Computer Vision*, 2009. 2

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016. 2

[5] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 1

[6] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conf. on Computer Vision*, pages 137–153, 2016. 1, 2, 3, 5, 8

[7] M. Jain, J. C. van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 740–747, 2014. 2

[8] M. Jain, J. C. van Gemert, and C. G. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 46–55, 2015. 2

[9] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 1, 2, 5

[10] D. Jurafsky, C. Wooters, J. Segal, A. Stolcke, E. Fosler, G. Tajchaman, and N. Morgan. Using a stochastic context-free grammar as a language model for speech recognition. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 1, pages 189–192, 1995. 5

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1, 2

[12] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 780–787, 2014. 1, 2, 5

[13] H. Kuehne, J. Gall, and T. Serre. An end-to-end generative framework for video segmentation and recognition. In *IEEE Winter Conf. on Applications of Computer Vision*, 2016. 2, 5

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *Int. Conf. on Computer Vision*, pages 2556–2563, 2011. 1

[15] H. Kuehne, A. Richard, and J. Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 2017. 1, 2, 3, 5, 7, 8

[16] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 2

[17] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017. 2

[18] C. Lea, A. Reiter, R. Vidal, and G. D. Hager. Segmental spatiotemporal CNNs for fine-grained action segmentation. In *European Conf. on Computer Vision*, pages 36–52, 2016. 1

[19] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What's cookin'? Interpreting cooking videos using text, speech and vision. In *Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015. 2

[20] P. Mettes, J. C. van Gemert, and C. G. M. Snoek. Spot on: Action localization from pointly-supervised proposals. In *European Conf. on Computer Vision*, 2016. 2

[21] D. Oneata, J. Verbeek, and C. Schmid. The LEAR submission at Thumos 2014. Technical report, Inria, 2014. 1, 2

[22] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 612–619, 2014. 2

[23] M. J. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67, 1994. 4

[24] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016. 1, 2, 3, 5

[25] A. Richard, H. Kuehne, and J. Gall. Weakly supervised action learning with RNN based fine-to-coarse modeling. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017. 1, 2, 3, 8

[26] A. Richard, H. Kuehne, A. Iqbal, and J. Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018. 2

[27] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1194–1201, 2012. 1, 2, 5

[28] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal on Computer Vision*, 119(3):346–373, 2016. 5

[29] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage CNNs. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016. 1, 2

[30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 1, 2

[31] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016. 2

[32] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1

[33] J. C. van Gemert, M. Jain, E. Gati, and C. G. Snoek. APT: Action localization proposals from dense trajectories. In *British Machine Vision Conference*, 2015. 2

[34] N. N. Vo and A. F. Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2641–2648, 2014. 2

[35] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Int. Conf. on Computer Vision*, pages 3551–3558, 2013. 1, 2, 5

[36] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmed-nets for weakly supervised action recognition and detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017. 3

[37] J. Wu, Y. Yu, C. Huang, and K. Yu. Deep multiple instance learning for image classification and auto-annotation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015. 6

[38] Y. Yan, C. Xu, D. Cai, and J. Corso. Weakly supervised actor-action segmentation via robust multi-task ranking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017. 2

[39] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016. 1, 2