Temporal Sequence Modeling for Video Event Detection

Yu Cheng, Quanfu Fan, Sharath Pankanti
IBM T.J. Watson Research Center
{yucheng,qfan,sharath}@us.ibm.com

Alok Choudhary
EECS Department, Northwestern University
choudhar@eecs.northwestern.edu

Abstract

We present a novel approach for event detection in video by temporal sequence modeling. Exploiting temporal information has lain at the core of many approaches for video analysis (i.e. action, activity and event recognition). Unlike previous works doing temporal modeling at semantic event level, we propose to model temporal dependencies in the data at sub-event level without using event annotations. This frees our model from ground truth and addresses several limitations in previous work on temporal modeling. Based on this idea, we represent a video by a sequence of visual words learnt from the video, and apply the Sequence Memoizer [21] to capture long-range dependencies in a temporal context in the visual sequence. This data-driven temporal model is further integrated with event classification for jointly performing segmentation and classification of events in a video. We demonstrate the efficacy of our approach on two challenging datasets for visual recognition.

1. Introduction

The exponential growth of video content today creates a great need for methods of intelligent video analysis and understanding. Among them, video event detection plays a central role in many applications such as surveillance, topic discovery and content retrieval. The task of event detection involves identifying the temporal range of an event in a video (i.e. *when*) and sometimes the location of the event as well (i.e. *where*). While there have been increasing efforts recently to tackle this problem, it remains rather challenging due to compounding issues such as large intra-variances of events, varied durations of events and the presence of background clutter.

In this work we aim to address the problem of video event detection by exploiting temporal dependencies among events. Realistic video events are often dependent, exhibiting short or long interactions between them depending on scenarios. As illustrated in Fig. 1, in an airport surveillance environment, *PeopleMeet* (a passenger approaching the information desk for direction) is followed by *Pointing* (the staff worker pointing to a direction) and then by *SplitUp* (the two people splitting up). More rich temporal patterns among events in an airport scenario are shown in Fig. 1.

Modeling temporal relationships and structures described above has lain at the core of human action, human activity and event recognition. Approaches for action recognition usually focus on capturing the underlying temporal structures of actions (i.e. *intra-dependencies*), either through feature representations [11] or using more sophisticated models [13, 8, 23]. In the meanwhile, works of activity recognition attempt to explore the temporal relationships between primitive actions of an activity (i.e. *inter-dependencies*) by graphical models such as HMMs and DBNs [7, 12, 17, 5].

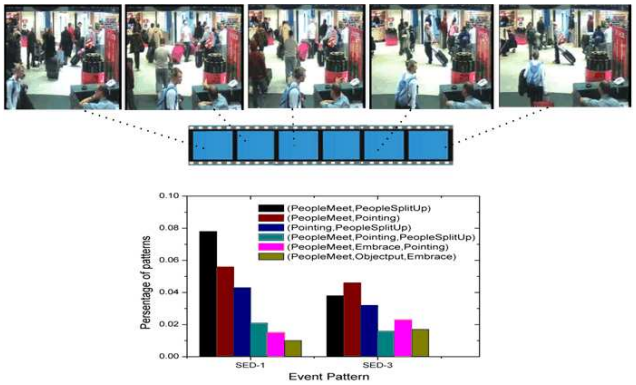

Figure 1. Temporal patterns exhibited on two cameras in the *SED* data [15]. The top images show such an example: *PeopleMeet* → *Pointing* → *SplitUp* (best view in zoom and color).

While temporal modeling has been enjoying great success in video understanding, for effective analysis of video events, several issues remains to be addressed. Firstly, the common practice of temporal modeling based on the 1^{st}-order Markov assumption can only capture a short interaction between the current and previous states. While this may suffice for human activity modeling where the temporal ordering of actions in an activity is well defined and often strict, it faces great limitations when applied to explore the relatively loose and sometimes long-range tem-

poral contexts (usually unknown) often presented in event data. The *n-gram* models such as [1] used in speech recognition might help, but in practice they tend to suffer from insufficient training data for capturing the complex relationships and computational scalability issues. Secondly, in many cases, there are only a few events of interest in a video and they are often accompanied with a substantially larger amount of *null* events or background clutter. Consequently, the Markov assumption will be heavily biased towards *null* events and weaken the dependencies between true events, leading to unsatisfactory performance. Thirdly, most approaches build temporal models directly from ground truth. Such models cannot discover temporal patterns associated with unannotated events regardless of how strong the dependencies are. For instance, the relationship between *PeopleMet*, *Pointing* and *SplitUp* in the above example, would not be captured if any two events were unavailable in the annotations.

To address the aforementioned limitations, we present a novel approach for video event detection based on temporal modeling. We formulate the detection task as a problem of sequence modeling where our goal is to break a visual sequence into segments of varied lengths and label them with events of interest or a *null* event. Based on this formulation,we first represent a video by a sequence of visual words learnt from our data in an unsupervised way with k-means clustering (Fig. 3). We then apply the Sequence Memoizer (SM) [21] to explore temporal dependencies among the visual words in the sequence. The SM, a non-parametric Bayesian approach initially developed for language modeling, can effectively model long-range contexts in discrete sequence data as well as the power-law properties [24] exhibited in a wide variety of problems. More specifically, SM-based sequence model is empowered with the ability to predict the occurrence of a subsequent visual word in a sequence conditioned on all its previous contexts observed. It is this ability that enables a robust way of temporal modeling without heavily relying on annotation. We finally integrate the sequence model and event classification into a framework that performs segmentation and classification of events jointly in a video.The optimal segmentation can be found efficiently by dynamic programming, similar to the work of [6].

An overview of our approach is illustrated in Fig. 2. To the best of our knowledge, this is one of the very few approaches that apply a viable statistical approach to model long-range contextual dependencies for a visual recognition problem. It presents several advantages over previous works. The sequence model is built upon visual words (sub-events), not on annotated events, thus it does not require ground truth. As demonstrated later, such temporal modeling on sub-event level is superior to its event-level counterpart. In addition, our approach automatically discovers the temporal contexts and structures inherent in the data and exploiting them to enhance detection. We validate our approach and demonstrate its efficacy using two challenging visual recognition datasets.

2. Related Work

A lot of schemes have been proposed for the human action recognition. Most of them perform classification on pre-segmented clips, exploiting temporal information either through feature representations [11, 20] or more sophisticated models [13, 8, 19, 23]. For example, Laptev et al. [11] applied bag of spatiotemporal interest points to classify human motion in realistic video settings. Tran & Sorokin [20] developed motion context features to learn nearest neighbor metric for classifying actions in YouTube videos. Niebles *et al.* [13] developed an unsupervised model for human actions detection based on probabilistic Latent Semantic Analysis. More recently, Tan et al. [19] developed a variant of HMM model that is trained in a max-margin framework to automatically discover discriminative and interesting segments of video. Zhang et al. [23] proposed an approach that can identify both local and long-range motion interactions to handle long-term activities more effectively.

For applications where the temporal range of an action or event needs to be identified in a video, sliding window is a popular technique [3, 9] to turn a classifier into a detection method. For example, Chen *et al.* [3] built an event classifier based on a Fisher vector coding representation for surveillance events, and then combined techniques of sliding windows, multi-scale detection and non-maximum suppression for event detection. The difficulty of such approaches is the determination of a classification threshold for true events. Due to this limitation, more recent efforts have proposed to learn framework for simultaneous segmentation and recognition in longer video sequences [18, 14, 6]. For example, Oh *et al.* [14] developed a linear dynamical system to model honeybee behavior. The work of [6] trains a discriminative recognition model with a multi-class SVM that maximizes the separating margin between classes, in a similar spirit of [18] which maximizes the overall classification scores.

Another related direction of our work is human activity recognition. Most of works in activity recognition explore the temporal relationships between primitive actions of an activity (i.e. *inter-dependencies*) using graphical models such as HMMs and DBNs [7, 12, 17, 5]. However, such approaches usually require domain knowledge to build or guide temporal modeling.

Our approach is different from these previous methods in several aspects: 1) our model can capture both intra-dependencies and inter-dependencies simultaneously by explicitly modeling the temporal relations over video segments; and 2) the model can capture and exploit long-range

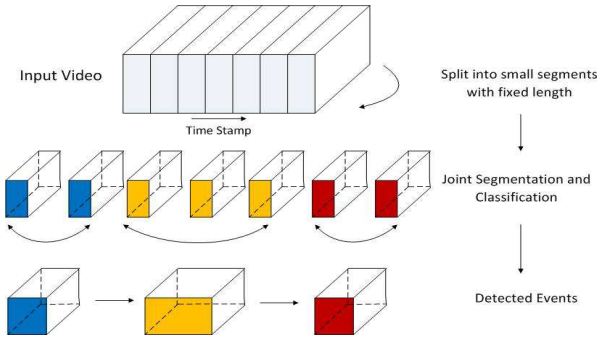

Figure 2. Given an input video, our approach divides it into a sequence of temporal segments uniformly and then builds a temporal model on top of the sequence to capture long contextual dependencies in the visual data sequence. Then the approach combines the temporal model and event classification to jointly perform event segmentation and classification.

temporal dependencies in the data; and 3) the model construction does not rely on event annotations or ground truth.

3. Our approach

Unlike most previous works on event detection such as [3] that treat video segmentation and event classification separately, our approach performs video segmentation and classification jointly with a temporal model described later in Section 4. The motivation behind temporal modeling is to exploit rich temporal structures and dependencies that often exist in event data to enhance detection. We start by introducing the video representation in our approach.

3.1. Video Representation

Given an input video $\mathbf{X}$, we first divide it into n temporal segments of a fixed length l_{seg}, i.e. $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$. We then compute the bag of words (BOW) feature for each segment upon motion SIFT key points [2]. The segments are further clustered into k visual words using k-means, and each segment is assigned a visual word. Finally, the video is represented by a sequence of visual words $\mathbf{W} = \{w_1, w_2, \cdots, w_n\}$. In our experiments, l_{seg} was set invariantly to the total length of the video, and k usually ranges from 600 to 900 depending on the complexity of the data.

Fig. 3 illustrates a few subsequences learnt from our data. One immediate observation is that the same event tend to generate similar visual words. A visual word from one event may statistically interact with another one from a different event, even though the two words can be temporally distant. For instance, K and P. We shall show how to model this type of long-range interactions later on in Section 4.

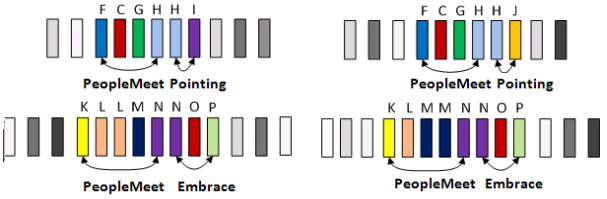

Figure 3. Samples of visual word sequences related to events. A same event tends to generate similar visual words. Words for *null* events are skipped here for clarity.

3.2. Joint Segmentation and Classification

With the video representation described above, our goal is to partition $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ into m units and label each unit with an event of interest or a *null* event (Fig. 2). Here a unit is a set of consecutive segments of $\mathbf{X}$. Let $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_m\}$ be such a partition where a unit $\mathbf{s}_i = \mathbf{X}_{t_i^1:t_i^2} = [\mathbf{x}_{t_i^1}, \ldots, \mathbf{x}_{t_i^2}]$ and t_i^1 and t_i^2 specify the start and end indices of the segments in $\mathbf{s}_i$. Also, let $\mathbf{Y} = \{y_1, y_2, \cdots, y_m\}$ where $y_i \in \mathbf{Y}$ is the event class label assigned to $\mathbf{s}_i$. To model temporal contexts in the data, we associate $\mathbf{S}$ with a visual sequence $\mathbf{Z} = \{z_1, z_2, \cdots, z_l\}$. The quality of the partition $\mathbf{S}$ with regard to event classification can then be evaluated by,

$$f(\mathbf{S}, \mathbf{Y}) = \sum_{i=1}^{m} \varphi(y_i|\mathbf{s_i}) + \mu \sum_{\substack{i=1 \\ 1 \le k \le i-1}}^{l} p(z_i|z_{i-k}, \cdots, z_{i-1}) \tag{1}$$

where μ is a trade-off parameter learnt from data empirically. Note that $\mathbf{Z}$ can be of any visual data sequence created on top of $\mathbf{S}$. For example, a sequence of visual events or visual words. We will further explain this in Section 4.2.

The first item $\varphi(y_i|\mathbf{s_i})$ in Eq. 1 measures the likelihood of the unit $\mathbf{s_i}$ being event y_i. We use the SVM classification score of $\mathbf{s_i}$ on event y_i for this item (see Section 5 for detail).

The second item $p(z_i|z_{i-k}, \cdots, z_{i-1})$ is provided by our sequence model discussed in Section 4.2. To put it simple, it is the probability of predicting z_i as the next symbol after seeing the previous k symbols from z_{i-k} to z_{i-1}. When $k = 1$, this item degrades to the well-studied 1^{st}-order Markov property. On the other hand, if $k = i - 1$, it puts the entire history of the sequence into consideration. Addressing such long contexts is a valid concern in previous work [22]. However, a recently developed probabilistic model [21] broke through this limitation by exploring an infinite length of context in a discrete data sequence.

Before detailing how to model the temporal sequence using the technique of [21] in the next section, we briefly describe how the above objective function can be solved efficiently by dynamic programming.

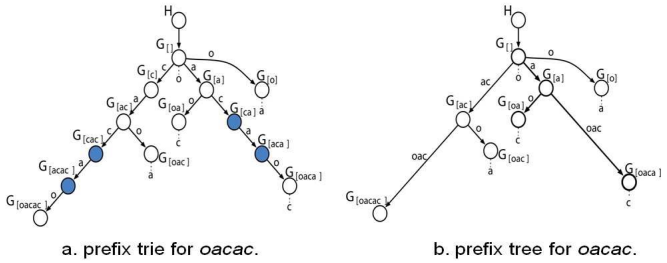

Figure 4. An example in [21] of prefix trie and prefix tree for the string *oacac*

3.3. Dynamic Programming

Performing segmentation on a new video sequence $\mathbf{X}$can be casted into the task of maximizing the objective function $f(\mathbf{S}, \mathbf{Y})$. Given any video flip $\mathbf{X}_{0:u}$ with length $u \in (0, n]$. let us consider a variation objective function $f(\mathbf{S}, \mathbf{Y}, u)$ according to u. Let $\mathbf{Z}_{u-l:u}$ be the visual sequence of $\mathbf{X}_{u-l:u}$ and $\mathbf{Z}_{0:u-l} = \{z_1, z_2, \cdots\}$ be the union of visual sequences backward. The transition function $\theta(u, l)$ can be expressed as:

$$\theta(u,l) = max(\varphi(y|\mathbf{X}_{u-l:u}) + \mu P(Z_{u-l:u}|\mathbf{Z}_{0:u-l}) \quad l_{min} \leq l \leq l_{max} \tag{2}$$

where y ranges all the possible event labels. $[l_{min}, l_{max}]$ gives the minimum and maximum durations of an event, which can be obtained from ground truth. The final task is to compute $f(\mathbf{S}, \mathbf{Y}, len(\mathbf{X}))$, which can be done by:

$$f(\mathbf{S}, \mathbf{Y}, u) = \underset{l_{min} \leq l \leq l_{max}}{\arg\max} \{\theta(u, l) + f(\mathbf{S}, \mathbf{Y}, u - l)\} \tag{3}$$

Unlike [6] which does exhaust search on each frame in the dynamic programming, our approach searches only on segments. The complexity of the implementation for segmentation on $\mathbf{X}$ is $O(m\frac{(l_{max}-l_{min}+1)}{l_{seg}} len(\mathbf{X}))$.

4. Temporal Modeling by Sequence Memoizer

To solve Eq. 1, we need to compute the probability of a visual label z_i conditioned on an observed sequence $\{z_{i-k}, \cdots, z_{i-1}\}$. We adopt the Sequence Memoizer (SM) [21] here for such a purpose.

4.1. Sequence Memoizer (SM)

Sequence Memoizer (A Stochastic Memoizer for Sequence Data) is an unbounded-depth, hierarchical, Bayesian nonparametric model of discrete sequences. Compared to other techniques for sequence modeling, the SM can more effectively learn a joint distribution over discrete sequences of flexible lengths and capture long-range dependencies. The approach has demonstrated state-of-the-art results for language modeling and data compression.

Given a sequence of discrete random variables $\mathbf{x_{1:T}} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$ of arbitrary length T, each taking values in a symbol set. The joint distribution over the sequence estimated by the SM is

$$p(\mathbf{x_{1:T}}) = \prod_{i=1}^{T} p(\mathbf{x}_i|\mathbf{x}_{1:i-1}) \tag{4}$$

which hints that each $\mathbf{x}_i$ is predicted given a context of all preceding variables $\mathbf{x}_{1:i-1}$. Note that this is different from an n^{th}-Markov assumption as T here can go to infinity theoretically.

The SM represents a sequence by a prefix trie (Fig.4.a), or a more efficient prefix tree (Fig. 4.b) that can be constructed from an input string in linear time and space complexity. Based on this representation, the SM places a Pitman-Yor prior (PYP) to approximate the frequency of each subsequence in the tree. This nicely addresses the problem of insufficient training data often encountered by traditonal sequence modeling based on n-th Markov assumption. Mathematically, the probability of $s \in \sum$ given its previous context $\mathbf{s}^{'}$, $G_{[\mathbf{s}]}$, is expressed by

$$G_{[\mathbf{s}]}|d_{[\mathbf{s}]}, c_{[\mathbf{s}]}, G_{[\mathbf{s}']} \sim \mathcal{PY}(d_{[\mathbf{s}]}, c_{[\mathbf{s}]}, G_{[\mathbf{s}']}), \tag{5}$$

where c and d are the parameters of the Pitman-Yor prior. $[\mathbf{s}] = [s\mathbf{s}^{'}]$. As shown in [21], using some special analytic marginalization technique, $G_{[\mathbf{s}]}$ can be computed efficiently in linear time. We refer the reader to [21] for further details.

In SM, the later symbols in a context are more important in predicting the subsequent symbol. Based on this idea, in the example illustrated in Fig. 3, the similarity between the two subsequences shown at the bottom (i.e. $\cdots KLLMNNOP\cdots$ and $\cdots KLMMNNOP\cdots$) is high as they share a long suffix. On the other hand, K is more important than others for predicting L as KL occurs in both of the two subsequences.

4.2. Temporal Sequence Modeling

A natural thought is to apply the SM to model an event sequence, similar to what an HMM does. This is straightforward and can be done easily by setting $\mathbf{z_i}$ in Eq. 1 to y_i directly. We call this method *event-level sequence modeling* (ESM). However, such a method, though being widely practiced, requires ground truth for model learning. As pointed out previously in Section 1, this model can not take full advantage of the SM due to event sparsity and extremely unbalanced distribution, and is also less robust in handling *null* events.

Realizing that Eq. 1 takes flexible visual sequences, we model the visual word sequence with the SM in Eq. 1. Such modeling at a granular level, referred to as segment-level sequence modeling (SSM) here, turns out to be more effective and robust in our experiments. This largely lies in that a) a

large number of visual words are more likely to present a power-law distribution than real events that are usually only a few; and b) the sequence model is constructed in a purely data-driven way, not from the event annotations.

We now show how to compute $p(z_i|z_1 \cdots z_{i-1})$ with the SM. Remember that the visual label z_i is associated with a unit $\mathbf{s}_i = [\mathbf{x}_{t_i^1}, \ldots, \mathbf{x}_{t_i^2}]$, which can be represented by a sequence of visual words $[w_{t_i^1}, \ldots, w_{t_i^2}]$ (See Section 3). By taking this into account and applying a chain rule, we can obtain,

$$\begin{aligned} p(z_i|z_{i-k} \cdots z_{i-1}) &= p(w_{t_i^1}, \ldots, w_{t_i^2}|w_{t_{i-k}^1}, \ldots, w_{t_{i-1}^2}) \\ &= \prod_{j=t_i^1}^{t_i^2} p(w_j|w_{t_{i-k}^1}, \ldots, w_{j-1}) \end{aligned} \tag{6}$$

By setting $k = i - 1$ in the above equation, the last item become $p(w_j|w_1, \ldots, w_{j-1})$. This can be computed by the SM efficiently.

5. Event Classification

In Eq. 1, we need to evaluate $\varphi(y_i|\mathbf{s_i})$ for any possible length l ($l_{min} \leq l \leq l_{max}$) of a unit $\mathbf{s_i}$ on all events including the *null* event class. We note that the temporal length of ground truth events can vary significantly. For example, the maximum length of a *PersonRuns* event is up to 1000 frames while the minimal length is only 10 frames. Such diversity in duration brings information loss if we learn a classification model with a single fixed temporal scale. We thus propose to learn classifiers on multiple temporal scales to match the initial video segmentation by a fixed length of l_{seg} frames described in Section 3.1.

Let $h = (l_{max} - l_{min})/l_{seg}$. Then it suffices to solve Eq. 1 if we train h classifiers for each event at each scale from l_{seg} to $h * l_{seg}$. In all our experiments, we used the same temporal range ($30 - 120$ frames) for all events and built 4 classifiers for each event at 30, 60, 90, and 120 frames, respectively, for efficiency. Note that this is consistent with the fact: $l = h * l_{seg}, (h = 2, 4, 6, 8), l_{seg} = 15$ and $l_{min} \leq l \leq l_{max}$. we use multi-class SVM [4] to train a model for each event class.

6. Experimental Results

We tested our proposed approaches on two challenging datasets: *Hollywood* [11] and TRECVID Surveillance Event Detection (*SED*) [15]. The former is a human action dataset retrieved from popular movies while the latter is a visual event detection dataset collected from a surveillance environment.

Figure 5. Typical video shots of the *SED* dataset. From left to right are *Pointing*, *CellToEar* and *PersonRuns* events.

6.1. Experimental Setup

We developed three sequence models based on the SM technique. The first one (*ESM*-∞), as described in Section 4.2, performs sequence modeling at event level, taking a full length of context into account. The second one (*ESM*-1), is a special case of the first one with only 1^{st}-order dependency considered, in a similar spirit of HMMs. The last one (*SSM*-∞) is what we propose, i.e. a segment-level sequence model exploiting a full length of context. These models are integrated into the framework described in Section 3 for event detection.

Baselines We implemented the approach proposed by Hoai *et. al.* in [6] and used it as the primary baseline in our evaluation. This approach performs joint segmentation and classification of human actions based on maximizing the margins of the top two event classification scores. However, it does not consider temporal relationships among events. On the *SED* dataset, in addition to Hoai's work, the approach developed by Chen *et. al* [3] was included in our comparison. While Chen's approach conducts event detection by sliding window, it has acheived state-of-the-art performance on the *SED* dataset, ranking on the top on 4 events out of 7 in the TRECVID SED 2012 evaluation.

Features for Classification We used STIP features [11] for event classification on *Hollywood*, and spatial-temporal Fisher Vector features [3] on *SED*. We adopted the Multi-class SVM method [4] to train a classifier for each event (action) plus a *null* event (action) class for all the approaches in comparison except Chen's, which does multiclassification using the one-against-all methodology. For each class, 4 classifiers were built at different temporal scales of 30, 60, 90 and 120 frames, respectively.

Event Detection and Evaluation To generate visual sequences for training our model *SSM*-∞, we used k-means to cluster a sequence of uniformly divided segments. On *Hollywood*, k was fixed to 200 in all the tests. On *SED*, we empirically determined k for each camera, which usually ranges between 600 and 900. A more detailed analysis on k is provided later in Section 6.4.

For each video in our evaluation, we first ran our approaches to find the optimal segmentation and class labels. At that point, each segment is assigned to a particular event class with a start and end frame. We then align the detection results with the ground truth (i.e the reference annotations) using a Bipartite matching method developed in [10]. If a

Events	Hoai[6]		*ESM*1		*ESM*-∞		*SSM*-∞	
	P	R	P	R	P	R	P	R
AnswerPhone	0.64	0.35	0.64	0.35	0.62	0.31	0.67	0.35
HugPerson	0.46	0.37	0.45	0.33	0.44	0.35	0.47	0.37
Kiss	0.44	0.49	0.43	0.51	0.43	0.49	0.44	0.49
SitDown	0.36	0.40	0.37	0.43	0.34	0.40	0.35	0.43
Overall	0.47	0.40	0.47	0.40	0.46	0.39	**0.48**	**0.41**

Table 1. Precisions (P) and Recalls (R) of different approaches on the new data set created from *Hollywood* (no temporal relationships among actions)

Events	Hoai[6]		*ESM*1		*ESM*-∞		*SSM*-∞	
	P	R	P	R	P	R	P	R
AnswerPhone	0.64	0.32	0.65	0.36	0.64	0.32	0.64	0.43
HugPerson	0.46	0.29	0.49	0.33	0.48	0.33	0.51	0.33
Kiss	0.40	0.48	0.42	0.52	0.42	0.52	0.44	0.59
SitDown	0.36	0.36	0.38	0.38	0.39	0.38	0.39	0.38
Overall	0.47	0.36	0.48	0.40	0.48	0.39	**0.50**	**0.42**

Table 2. Precisions (P) and Recalls (R) of different approaches on the new data set created from *Hollywood* (with enforced temporal relationships among actions)

detection is matched to a true event, it is considered a true positive. otherwise it is a false positive.

6.2. Evaluation on Hollywood Dataset

Data *Hollywood* is a video dataset focusing on realistic human actions. These actions include *AnswerPhone, HugPerson, Kiss, SitDown, SitUp, GetOutCar, HandShake,* and *StandUp*. This dataset is divided into two disjoint subsets with 219 video samples in the training set and 211 in the test set, respectively. Following [6], we selected the first four classes as actions to be recognized, and treated the others as *null* class.

Since *Hollywood* contains only pre-segmented clips, we created new video clips of longer durations for our evaluation purpose, by concatenating video clips picked from the original dataset. Two such datasets were created, both using all the clips from the training and testing sets. In the first one the clips were selected in an random order for concatenation. In the second one, in order to enforce temporal relationships, some clips were selected to exhibit temporal dependency. Specifically, we inserted some actions with 1-order dependency (such as *SitDown-AnswerPhone*) and 2-order dependency (such as *HugPerson-Kiss-SitDown*) to the data. A total of about 40 such video samples were formed in such a way, half into the training set and half into the testing set.

Results We reported the results on *Hollywood* by standard precision and recall metrics. As shown in Table 1, when there are no temporal relationships among events in the data, all approaches perform similarly, with our proposed approach (*SSM*-∞) doing slightly better than others. However, when temporal dependencies were added to events in the data, all the approaches with temporal modeling outperform the baseline, suggesting that temporal information is helpful for event detection. As expected, *SSM*-∞ achieves the best results in terms of both precision and recall, demonstrating a large improvement over the baseline. There is not much difference between *ESM*-1 and *ESM*-∞ in this test, because the datasets were contrived to exhibit only simple temporal relationships and the scenes in the clips differ significantly from movie to movie, leaving little long temporal context for exploitation.

6.3. Evaluation on SED Dataset

The SED dataset was captured from 5 surveillance cameras at different locations in a busy airport. The dataset has been used in the TRECVID SED evaluation track since 2009 to support the development of technologies of visual event detection in a large collection of streaming video data. It contains 10 surveillance events with people engaged in particular activities. Among them, 7 events were used in the TRECVID 2012 evaluation, including *CellToEar*, *Embrace*, *ObjectPut*, *Pointing*, *peopleMeet*, *PeopleSplitUp* and *PersonRuns*. This is an extremely challenging dataset for event detection, due to many confounding issues such as high-level activity, camera view changes, large variances of events (i.e. PeopleMeet) and small objects (i.e. CellToEar) (Fig. 5). The annotations of *SED* only include temporal extents and event labels.

We used the development set of *SED* (about 100 hours of video data) for our evaluation. The data were split into two equal parts for training and testing.

Results The results of different approaches on *SED* are listed in Table 3. In addition to precision and recall, the scores of the Detection Cost Rate (DCR), a performance metric adopted in the TRECVID evaluation [16], are provided in Table 3. Basically, DCR is a linear combination of two errors: Missed Detections and False Alarms. It reflects a tradeoff between these two types of errors by weighing them differently in scoring. *A lower DCR indicates a better performance*. More details about this metric can be found in [16].

First, we observe that while Hoai's approach performs reasonably well on *Hollywood* against other approaches, it fails to yield comparable results on *SED*. Also, it has very low recalls for *Cell2Ear*, *ObjectPut* and *Pointing* events, the most difficult ones to detect in this dataset. On the other hand, with the help of temporal information, *ESM*-1 already achieves similar performance to Chen's approach. By further exploring longer temporal contexts, *ESM*-∞ and *SSM*-∞ outperform Chen's approach, clearly demonstrating the benefit of modeling more complex temporal interactions in the sequence data. *SSM*-∞ produces the best results on all the events except on *Cell2Ear*. Some of the difficult events such as *Pointing* and *ObjectPut* have seen significant improvements in *SSM*-∞ over the baselines, indicating the efficacy of temporal modeling by our approach. We also no-

Events	#Ground truth	Chen[3]			Hoai[6]			*ESM*1			*ESM*-∞			*SSM*-∞		
		P	R	DCR	P	R	DCR	P	R	DCR	P	R	DCR	P	R	DCR
Cell2Ear	374	0.21	0.06	0.953	0.13	0.01	1.002	0.30	0.06	0.953	0.34	0.07	**0.933**	0.28	0.05	0.95
Embrace	479	0.13	0.27	0.835	0.12	0.24	0.856	0.14	0.27	0.833	0.15	0.28	0.812	0.17	0.34	**0.764**
ObjectPut	1898	0.33	0.05	0.985	0.43	0.01	1.001	0.39	0.05	0.969	0.44	0.06	0.941	0.45	0.09	**0.918**
PeopleMeet	1376	0.19	0.27	0.931	0.17	0.26	0.942	0.19	0.27	0.933	0.19	0.28	0.919	0.20	0.28	**0.913**
SplitUp	762	0.20	0.37	0.819	0.17	0.32	0.897	0.21	0.36	0.767	0.21	0.36	0.764	0.22	0.38	**0.715**
PersonRun	365	0.21	0.52	0.573	0.19	0.44	0.761	0.20	0.51	0.569	0.20	0.52	0.564	0.23	0.59	**0.499**
Pointing	2338	0.21	0.12	1.009	0.20	0.03	1.018	0.21	0.11	0.998	0.22	0.13	0.983	0.26	0.19	**0.958**
Overall	7592	0.19	0.15	N/A	0.20	0.17	N/A	0.20	0.19	N/A	0.24	0.22	N/A	**0.25**	**0.27**	N/A

Table 3. Precisions(P), Recalls(R) and DCRs of different approaches on *SED*. Note that a lower DCR score indicates a better performance. The overall DCR performance is not available as the evaluation tool provided by TRECVID only outputs a score for each individual event.

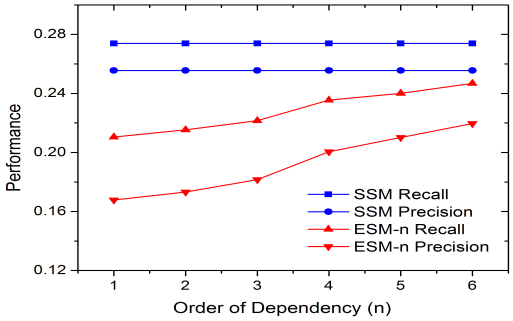

Figure 6. The performance (precision and recall) comparison using different lengths of temporal contexts.

events	*ESM*-∞			*SSM*-∞		
	P	R	DCR	P	R	DCR
Pointing	0.27	0.12	1.004	0.36	0.20	0.950
PersonRun	0.11	0.20	0.806	0.15	0.30	0.785
ObjectPut	0.39	0.06	0.944	0.45	0.09	0.933
Embrace	0.18	0.22	0.809	0.2	0.33	0.785

Table 4. performance of *ESM*-∞ and *SSM*-∞ based on only partial ground truth (only on Camera SED-1)

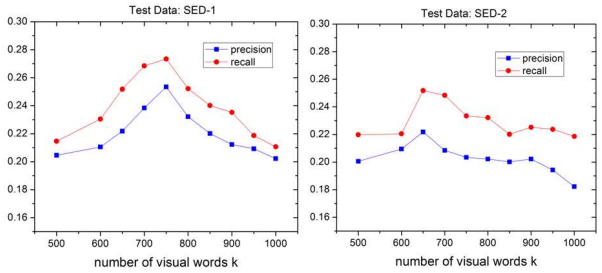

Figure 7. Performance of our approach varying by the number of visual words on two subsets (SED-1 and SED-2)

tice that our temporal modeling tend to have little effect on those events exhibiting no evident temporal dependencies on other events, such as *Cell2Ear* and *PeopleRun*.

6.4. Discussions

Below we provide more detailed analysis of our approach to support our main claims in this paper.

Long-range Temporal Dependencies. To demonstrate the efficacy of exploiting long-range temporal dependencies in modeling, we compared the performance of our proposed approach with the n-gram temporal model [22]. Fig 6 clearly illustrates the benefit of modeling long-range temporal contexts in the data. As the range of temporal dependencies increases, the performance improves consistently.

Effects of Ground Truth. To further understand the effects of ground truth in temporal modeling, we designed an experiment by comparing the results of our approaches using the original ground truth set and a reduced one. Specifically, we took out two events (i.e. *PeopleMeet* and *SplitUp*) from the event annotations on Camera 1 in *SED*, and ran *ESM*-∞ and *SSM*-∞ with the modified annotations. Note that these two masked out events have significant contributions to the temporal patterns in *SED*. From Table 4, we can see that *ESM*-∞ cannot stand up to expectation without the temporal relationships explicitly indicated in the ground truth. In comparison, *SSM*-∞ has not been affected much by the imperfect annotations. This experiment strongly supports our claim that at sub-event level granularity it is much more effective for the SM to capture the temporal dependencies.

Sensitivity Analysis K-means clustering is used to generate visual sequences for our modeling. Choosing a proper k can help discover fine-detailed temporal structures in the data. Generally speaking, k has to do with the complexity of the data. The more complex the scene is, a larger k is expected. In our experiments we empirically determined the number of visual words on each camera. To better understand how k affects the performance, we assessed the sensitivity of performance with respect to k. As illustrated in Fig 7, while some careful tuning of k is desirable for better performance, choosing a k between 600 and 800 on the *SED* dataset can yield reasonably good performance.

7. Conclusions

In this paper we have proposed a joint-segmentation-detection framework with temporal dependencies among events considered to enhance detection in videos. The dependencies are learned on visual word sequences using Squence Memoizer, which can capture long range dependencies and power-law characteristics. In addition, our model is constructed without relying on event annotations and is capable to handle *null* events well. We have shown

competitive results on difficult datasets and demonstrated that our approach outperforms state-of-the-art event detection methods.

Additionally, note that we undertook only limited joint segmentation (e.g., overlap between true events and correctly detected events) and recognition error analysis (false positives/negatives) in one corpus of the dataset (*SED*) when the domain experts (NIST and governmental authorities) specified the relative weights of the individual errors in the form of DCR metric [16]. In near future, we will more comprehensively explore the performance trade-offs between localization and categorization.

References

[1] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[2] M.-y. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. 2009.

[3] Q. Chen, Y. Cai, L. Brown, A. Datta, Q. Fan, R. Feris, S. Yan, A. Hauptmann, and S. Pankanti. Spatio-temporal fisher vector coding for surveillance event detection. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 589–592. ACM, 2013.

[4] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[5] Q. Fan, R. Bobbitt, Y. Zhai, A. Yanagawa, S. Pankanti, and A. Hampapur. Recognition of repetitive sequential human activity. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 943–950. IEEE, 2009.

[6] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3265–3272. IEEE, 2011.

[7] S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden markov models. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1455–1462. IEEE, 2003.

[8] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1165–1172. IEEE, 2009.

[9] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 166–173. IEEE, 2005.

[10] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[12] B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[13] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[14] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *Int. J. Comput. Vision*, 77(1-3):103–124, May 2008.

[15] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[16] P. Over, G. M. Awad, J. Fiscus, B. Antonishek, M. Michel, A. F. Smeaton, W. Kraaij, and G. Quénot. Trecvid 2010–an overview of the goals, tasks, data, evaluation mechanisms, and metrics. 2011.

[17] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1593–1600. IEEE, 2009.

[18] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[19] K. Tang, F. Li, and K. Daphne. Learning latent temporal structure for complex event detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1250–1257. IEEE, 2012.

[20] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 548–561, Berlin, Heidelberg, 2008. Springer-Verlag.

[21] F. Wood, C. Archambeau, J. Gasthaus, L. James, and Y. W. Teh. A stochastic memoizer for sequence data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1129–1136. ACM, 2009.

[22] F. Wood, J. Gasthaus, C. Archambeau, L. James, and Y. W. Teh. The sequence memoizer. *Communications of the ACM*, 54(2):91–98, 2011.

[23] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spatio-temporal phrases for activity recognition. In *Computer Vision–ECCV 2012*, pages 707–721. Springer, 2012.

[24] G. K. Zipf. *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press, 1932.