Journal of Artificial Intelligence Research 52 (2015) pages Submitted 08/14; published 03/15

A Compositional Framework for Grounding Language Inference, Generation, and Acquisition in Video

Haonan Yu HAONAN@HAONANYU.COM
N. Siddharth SIDDHARTH@IFFSID.COM
Andrei Barbu ANDREI@0XAB.COM
Jeffrey Mark Siskind QOBI@PURDUE.EDU
School of Electrical and Computer Engineering
Purdue University
465 Northwestern Avenue
West Lafayette, IN 47907-2035 USA

Abstract

We present an approach to simultaneously reasoning about a video clip and an entire natural-language sentence. The compositional nature of language is exploited to construct models which represent the meanings of entire sentences composed out of the meanings of the words in those sentences mediated by a grammar that encodes the predicate-argument relations. We demonstrate that these models faithfully represent the meanings of sentences and are sensitive to how the roles played by participants (nouns), their characteristics (adjectives), the actions performed (verbs), the manner of such actions (adverbs), and changing spatial relations between participants (prepositions) affect the meaning of a sentence and how it is grounded in video. We exploit this methodology in three ways. In the first, a video clip along with a sentence are taken as input and the participants in the event described by the sentence are highlighted, even when the clip depicts multiple similar simultaneous events. In the second, a video clip is taken as input without a sentence and a sentence is generated that describes an event in that clip. In the third, a corpus of video clips is paired with sentences which describe some of the events in those clips and the meanings of the words in those sentences are learned. We learn these meanings without needing to specify which attribute of the video clips each word in a given sentence refers to. The learned meaning representations are shown to be intelligible to humans.

1. Introduction

People use their knowledge of language to make sense of the world around them, not just to describe their observations or communicate to others. In this work, we present an approach which is able to describe video clips in natural language while simultaneously using that capacity to reason about the content of those clips. While earlier approaches can detect individual features in video (Laptev, 2005; Kuehne, Jhuang, Garrote, Poggio, & Serre, 2011), such as objects or events, we show how knowledge of language can integrate information from these different feature detectors in order to both improve their performance and support novel functionality. To do this, we exploit the compositional nature of language to construct models for entire sentences from individual word models, and use such models to determine if an entire sentence describes a video clip. We call the mechanism for determining how well a video clip depicts a sentence, and alternatively how well a sentence describes a

video clip, the *sentence tracker* (Yu & Siskind, 2013; Siddharth, Barbu, & Siskind, 2014), because it simultaneously performs multi-object tracking and recognition of events described by sentences. This ability to score video-sentence pairs also allows us to perform another important task that humans naturally engage in: learning word meanings. We show how the sentence tracker can perform this task using the same kind of information that is available to children, namely, video paired with entire sentences which describe some of the events depicted. This general-purpose inference mechanism for combining bottom-up information from low-level video-feature detectors and top-down information from natural-language semantics allows us to perform three novel tasks: tracking objects which are engaged in a specific event as described by a sentence, generating a sentence to describe a video clip, and learning word meaning from video clips paired with entire sentences.

Fundamentally, our approach relies on solving two separate problems simultaneously: tracking the participants of an event and recognizing the occurrence of that event. We formulate this as the combination of two measures: a measure of how well a video clip depicts a track collection and how well that track collection depicts an event. Note that what we mean by 'event' is a complex state of affairs described by an entire sentence, not the common definition used in the computer vision community, which refers to a single verb label attached to a video clip. In order to solve both problems simultaneously, we show how the similarity between tracking and event recognition facilities a common inference algorithm. We perform single-object tracking by combining the output of an unreliable detection source, an object detector, with an estimate of the motion present in the video, optical flow. The tracks produced consist of strong detections and their motion agrees with the motion present in the video. We perform single-word recognition by representing the meaning of a word in terms of the gross motion of object tracks. Finally, we show how single-object tracking and single-word recognition combine to perform multi-object tracking and whole-sentence recognition by exploiting the compositionality of language to combine word models into sentence models and by formulating both tasks in a way that is amenable to dynamic programming.

This ability to perform both tasks simultaneously—in other words, to score a video-sentence pair with how well the video clip depicts the sentence—is crucial for attaining good performance. By integrating top-down and bottom-up information, it corrects errors in object-detector output. This is important because object detectors are highly unreliable, achieving at most 40%-50% average precision on the PASCAL Visual Object Classes (VOC) challenge (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010). Barbu, Siddharth, Michaux, and Siskind (2012b) showed how the reliability of object tracking and single-word recognition (typically for a verb) can be improved by performing both simultaneously. We build on this earlier work and extend it to track multiple objects and recognize whole sentences. We further extend that work with a novel approach to sentence generation and learning word meanings.

Following Yamoto, Ohya, and Ishii (1992), Siskind and Morris (1996), and Starner, Weaver, and Pentland (1998), we represent word meanings in a fashion that can be grounded in video as multi-state time-series classifiers, either hidden Markov models (HMMs) or finite-state machines (FSMs), over features extracted from object tracks in such video. For example, a model for *approach* might use three states to encode an event where the distance between two tracked objects is initially high, over time decreases, and finally ends

by being small. Those earlier approaches confined themselves to representing the meaning of verbs, but we employ the same representation for all words in the lexicon regardless of their part of speech. This allows us to combine word models together into sentence models, in essence, creating large factorial models. Unlike earlier work (Kulkarni, Premraj, Dhar, Li, Choi, Berg, & Berg, 2011; Hanckmann, Schutte, & Burghouts, 2012; Barbu, Bridge, Burchill, Coroian, Dickinson, Fidler, Michaux, Mussman, Siddharth, Salvi, Schmidt, Shangguan, Siskind, Waggoner, Wang, Wei, Yin, & Zhang, 2012a; Krishnamoorthy, Malkarnenkar, Mooney, Saenko, & Guadarrama, 2013), we exploit linguistics, namely the concept of linking, to construct the particular factorial model which encodes the predicate-argument structure of a specific sentence, not all sentences which happen to share the same words. For example the sentence, *The person picked up the backpack* has very different meaning from the sentence *The backpack picked up the person*, despite sharing all words, and our method encodes such distinctions.

An overview of the operation of the sentence tracker is shown in Figure 1. Information is extracted from video using object detectors and optical flow, as discussed in Section 2.1. Independently, a sentence is parsed and the number of participants is determined, together with a linking function, as discussed in Sections 3. Each word in the sentence has an associated model, as discussed in Section 2.2. The information extracted from the sentence combines with the per-word models to form a model for an entire sentence, as discussed in Sections 2.3 and 2.4. That model takes, as input, the data extracted from a video clip and computes how well the clip depicts the given sentence, the video-sentence score shown in Equation 10.

In order to more formally articulate this approach and its applications, we represent the measure of how well a video clip depicts a sentence as a function $\mathcal{S} : (\mathbf{B}, \mathbf{s}, \Lambda) \mapsto (\tau, \mathbf{J})$, where $\mathbf{B}$ represents the information extracted from a video clip, $\mathbf{s}$ represents the sentence, Λ represents word meanings, τ is the video-sentence score, and $\mathbf{J}$ is a collection of tracks, one for each participant in the event described by the sentence, corresponding to the optimal video-sentence score. We use $\mathcal{S}_\tau$ and $\mathcal{S}_\mathbf{J}$ to refer to the two components produced by $\mathcal{S}$. This function internally makes use of the number L of event participants and θ, a *linking function*. The linking function maps arguments of words in the sentence to event participants. We make use of a *linking process*, a function $\Theta : \mathbf{s} \mapsto (L, \theta)$, that will be described in Section 3, to derive the number L of participants and the linking function θ. We now elaborate on three applications of this approach that we will demonstrate: *language inference*, *language generation*, and *language acquisition*.

In *language inference*, one can apply the sentence tracker to the same video clip $\mathbf{B}$, that depicts multiple simultaneous events taking place in the field of view, with two different sentences $\mathbf{s}_1$ and $\mathbf{s}_2$. In other words, one can compute $\mathbf{J}_1 = \mathcal{S}_\mathbf{J}(\mathbf{B}, \mathbf{s}_1, \Lambda)$ and $\mathbf{J}_2 = \mathcal{S}_\mathbf{J}(\mathbf{B}, \mathbf{s}_2, \Lambda)$ to yield two different track collections $\mathbf{J}_1$ and $\mathbf{J}_2$ corresponding to the different sets of participants in the different events described by $\mathbf{s}_1$ and $\mathbf{s}_2$. We demonstrate this in Section 5.3. Specifically, we show how language inference, unlike many other approaches to event recognition, not only deals with video that depicts multiple simultaneous events, but is also sensitive to subtle changes in sentence meaning. We present an experiment where we construct minimal pairs of sentences, given a grammar, which differ in only a single lexical constituent, where that varying lexical constituent can itself vary among all parts of speech and sentential positions. For example the two sentences

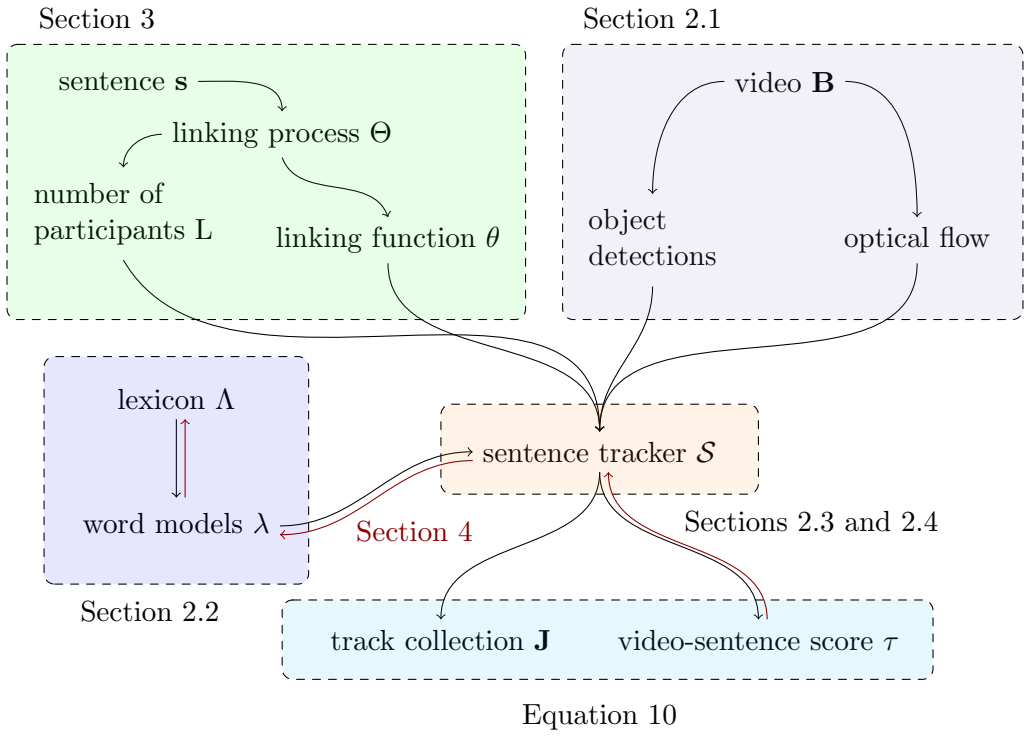


Figure 1: An overview of the approach presented and a roadmap to its presentation. Section 5.3 demonstrates language inference. Section 5.4 demonstrates language generation. Section 5.5 demonstrates language acquisition.

> *The person to the left of the trash can put down an object.*
> *The person to the right of the trash can put down an object.*

are minimal pairs which differ in the preposition attached to the subject noun phrase. We construct a video corpus where both sentences in such minimal pairs occur simultaneously in the same video clip and demonstrate how language inference is sensitive to changes in sentential meaning by producing two distinct and semantically appropriate sets of tracks given each of the two sentences as input. To conduct a thorough[1] evaluation, we employ a vocabulary of 17 lexical items (5 nouns, 2 adjectives, 4 verbs, 2 adverbs, 2 spatial-relation prepositions, and 2 motion prepositions) and a video corpus of 94 clips.

1. By 'thorough' we mean the following:
   1. We evaluate all three of the applications of our general method: inference, generation, and acquisition.
   2. We show performance on our entire corpus, without cherry picking.
   3. We illustrate deep semantic grounding by way of minimal pairs that vary all lexical items and all sentential positions.
   4. We demonstrate deep semantic grounding by rendering the thematic-role assignments for sentences on associated videos, illustrating correct assignment of event participants to roles and predicate arguments.
   5. We compare our learned models with ground-truth meaning representations and precisely measure the KL divergence of such in Table 10.

In *language generation*, we take a video clip $\mathbf{B}$ as input and systematically search the space of all possible sentences $\mathbf{s}$, that can be generated by a context-free grammar, and find the sentence with maximal video-sentence score:

$$\underset{\mathbf{s}}{\operatorname{argmax}}\, \mathcal{S}_\tau(\mathbf{B}, \mathbf{s}, \Lambda)$$

This generates a sentence that best describes an input video clip $\mathbf{B}$. We demonstrate this in Section 5.4. Unlike previous approaches to sentence generation from video which are largely *ad hoc*(Barbu et al., 2012a; Hanckmann et al., 2012; Krishnamoorthy et al., 2013), we present an approach which is optimal, in the sense that the generated sentence is that which will produce the highest video-sentence score. Our evaluation for language generation uses the same video corpus, grammar, and lexicon as used for language inference.

In *language acquisition*, we exploit the fact that we can simultaneously reason both about the presence and motion of participants in a video clip and about the meaning of a sentence describing that clip to compute models for word meaning from a training set of video clips paired with sentences. In other words, given a training set $\{(\mathbf{B}_1, \mathbf{s}_1), \ldots, (\mathbf{B}_M, \mathbf{s}_M)\}$ of video-sentence pairs where the word meanings Λ are unknown, we compute

$$\underset{\Lambda}{\operatorname{argmax}} \sum_{m=1}^{M} \mathcal{S}_\tau(\mathbf{B}_m, \mathbf{s}_m, \Lambda)$$

which finds the word meanings Λ that maximize the aggregate score for all video-sentence pairs in the training set. We demonstrate this in Section 5.5. We learn word meanings without needing to annotate which word refers to which attribute of the video and without annotating the tracks for the objects which participate in the event described in the training sentences. To conduct a thorough evaluation, we employ a vocabulary of 16 lexical items (6 nouns, 4 verbs, 2 adverbs, 2 spatial-relation prepositions, and 2 motion prepositions) and a video corpus of 94 clips out of which a total of 276 video-sentence pairs are constructed.

The central contribution of this work is the sentence tracker, a precise mathematical and computational framework for performing simultaneous object detection, multi-object tracking, action recognition, and recognition of multiple predicates assigned to different subsets of participants, culminating in Equation 10, as implemented as an efficient algorithm as illustrated in Figures 11 and 12, that implements exact inference in a joint model, along with the method for training such solely from videos paired with sentential annotation. The current focus in the computational linguistics community has been on large-scale unrestricted text processing for a long time now. The computer vision community is currently undergoing a similar transition towards processing large-scale unrestricted image and video corpora. Our sentence tracker is *not* intended to process unrestricted text and video. Nor is it intended to produce natural-sounding text descriptions of video. We are more concerned with semantics, as reflected by the truth of the text descriptions and the accuracy of the learned meaning representations. Moreover, we evaluate it on a corpus that is considerably smaller than what is currently used in both the computational linguistics and computer vision communities. We do this because we intend our work to address an orthogonal set of concerns:

1. We provide a unified framework that supports inference, generation, and acquisition.

2. We demonstrate that it learns correct meanings of all words, with no prior meanings of any words, from video paired with whole sentences, with no manual guidance as to which words correspond to which components of the video.
3. We demonstrate that it has deep understanding of sentential semantics, grounded in video, and that such are derived by a systematic computational process from deep word meanings grounded in video.
4. This deep understanding allows the framework to distinguish between subtle semantic distinctions that are manifest in two sentences that differ in a single word or in word order, *i.e.,* it understands the mapping between objects detected in the video and the particular semantic roles they play in the sentences.

This does not mean that it has greater or lesser limitations than current work in computational linguistics or computer vision. Different research has different limitations. The above four points highlight some of the limitations that such current work exhibits that are absent from the work presented here.

2. Joint Tracking and Event Recognition

We represent word meanings, and ultimately sentence meanings, as constraints over the time-varying spatial relations between event participants: their relative and/or absolute positions, velocities, and/or accelerations. This requires that we track the positions of event participants over the course of a video clip. In an ideal world, we would be able to accurately determine which object classes were present in any video frame and for those that are, precisely determine the positions of all instances of those classes in the field of view. Unfortunately, the current state of the art in object detection is far from this ideal. Object detectors only achieve between 3.8% and 65% average precision on the PASCAL VOC benchmark (Everingham et al., 2010). This means that, in practice, they suffer from both false positives and false negatives, as illustrated in Figure 2. While we wish to produce a single detection for each of the person and backpack, as shown in Figure 2(a), in practice, we often get spurious detections (false positives), as happens for the person detector in Figure 2(b), and fail to obtain the desired detection (false negatives), as happens for the backpack detector in Figure 2(c).

2.1 Detection-Based Tracking

The general approach to resolving this problem is to *overgenerate*. We lower the acceptance threshold for the detector, trading off a higher false-positive rate for a lower false-negative rate, as in Figure 2(d). We attempt to lower the threshold sufficiently to completely eliminate false negatives, biasing it to have a preponderance of false positives. The tracking problem then reduces to the problem of selecting detections from the frames of a video clip to assemble coherent tracks.

Let us assume, for a moment, that we wish to track a single instance of a specified object class known to be present in the field of view throughout a video clip. We track that object by selecting a single detection in each frame from the pool of detections for that object class. The sequence of the top-scoring detection in each frame might not be temporally coherent, as shown in Figure 3(a). Likewise, the most temporally-coherent sequence of detections might consist of low-scoring misdetections, as shown in Figure 3(b). Thus our

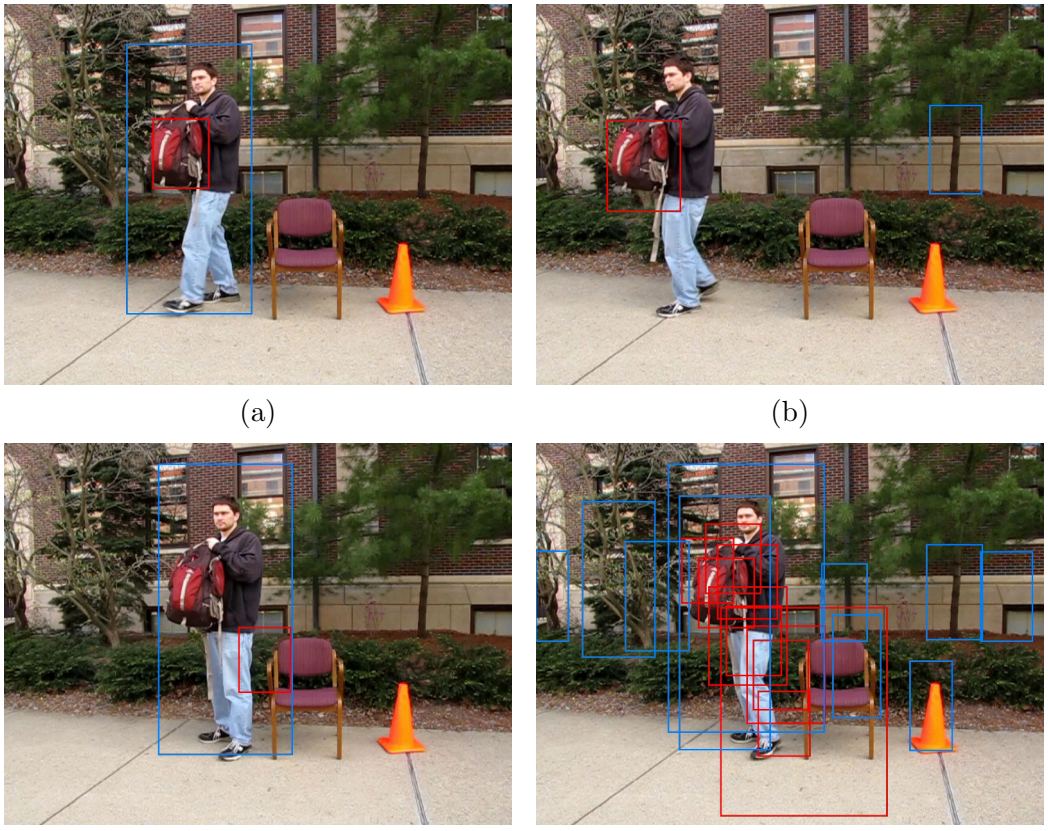


Figure 2: State-of-the-art object detectors are imperfect. While we wish a single detection for the person and backpack, as in (a), in practice we often get spurious detections (false positives), as in (b), or fail to obtain the desired detection (false negatives), as in (c). Reducing the acceptance threshold biases the detector to trade off a higher false-positive rate for a lower false-negative rate, as in (d).

approach is to balance these two extremes by incorporating both the detection score and a temporal-coherence score into the selection criterion. This often can yield the desired track, as shown in Figure 3(c).

We adopt an objective function that linearly combines both the sum of the detection scores in all video frames and the sum of a temporal-coherence score applied to all pairs of adjacent video frames. More formally, in a video clip $\mathbf{B}$ of T frames, with J^t detections $b_1^t, \ldots, b_{J^t}^t$ in frame t, we seek a track $\mathbf{j}$, namely a sequence $j^1, \ldots, j^T$ of detection indices, that maximizes the sum of the detection scores $f(b_{j^t}^t)$ and the temporal-coherence scores $g(b_{j^{t-1}}^{t-1}, b_{j^t}^t)$:

$$\max_{\mathbf{j}} \left(\sum_{t=1}^{T} f(b_{j^t}^t) \right) + \left(\sum_{t=2}^{T} g(b_{j^{t-1}}^{t-1}, b_{j^t}^t) \right) \tag{1}$$

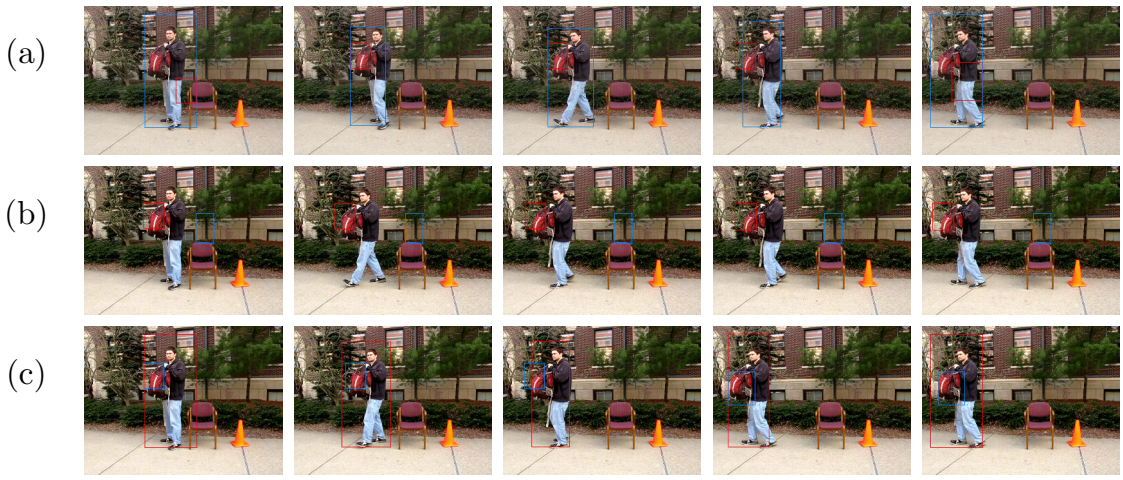


Figure 3: Assembling a track from a single detection per frame selected from a pool of overgenerated detections. Selecting the top-scoring detection in each frame of a video clip can yield an incoherent track, as shown in (a). Selecting tracks to maximize temporal coherence can lead to tracks incorporating solely low-scoring misdetections, as shown in (b). Selecting tracks to maximize an appropriate combination of detection score and temporal-coherence score can lead to the desired track, as shown in (c).

The objective function in Equation 1 constitutes a measure of how well a video clip $\mathbf{B}$ depicts a track $\mathbf{j}$. We employ this particular objective function because it can be optimized efficiently with dynamic programming (Bellman, 1957), namely the Viterbi (1967) algorithm. This leads to a lattice, as shown in Figure 4. The columns of the lattice correspond to video frames, the detections in each frame constitute the columns, and a track constitutes a path through the lattice.

The general approach to tracking by overgenerating detections and selecting among those to yield a track is known as *detection-based tracking* (Han, Sethi, Hua, & Gong, 2004; Avidan, 2004; Wu & Nevatia, 2007). Our approach to using the Viterbi algorithm for this purpose was first explored by Wolf, Viterbi, and Dixon (1989) to track radar detections. It relies on an analogy:

> "... detections correspond to HMM states, the detection score corresponds to the HMM output probability, the temporal-coherence score corresponds to the HMM state-transition probability, and finding the optimal track corresponds to finding the maximum *a posteriori* probability (MAP) estimate of the HMM state sequence (where the computation of the MAP estimate is performed in log space)."

We crucially rely on this analogy for the entire remainder of this paper.

The above can trivially be modified to denote a MAP estimate in log space with suitable normalization by a constant factor. For our purposes, however, all that is relevant is that it optimizes a linear combination of two score components: the sum of state-based scores and the sum of transition-based scores. In particular, the Viterbi algorithm can be applied to Equation 1, without any constraint on permissible values for the scores $f(b)$ and $g(b', b)$.

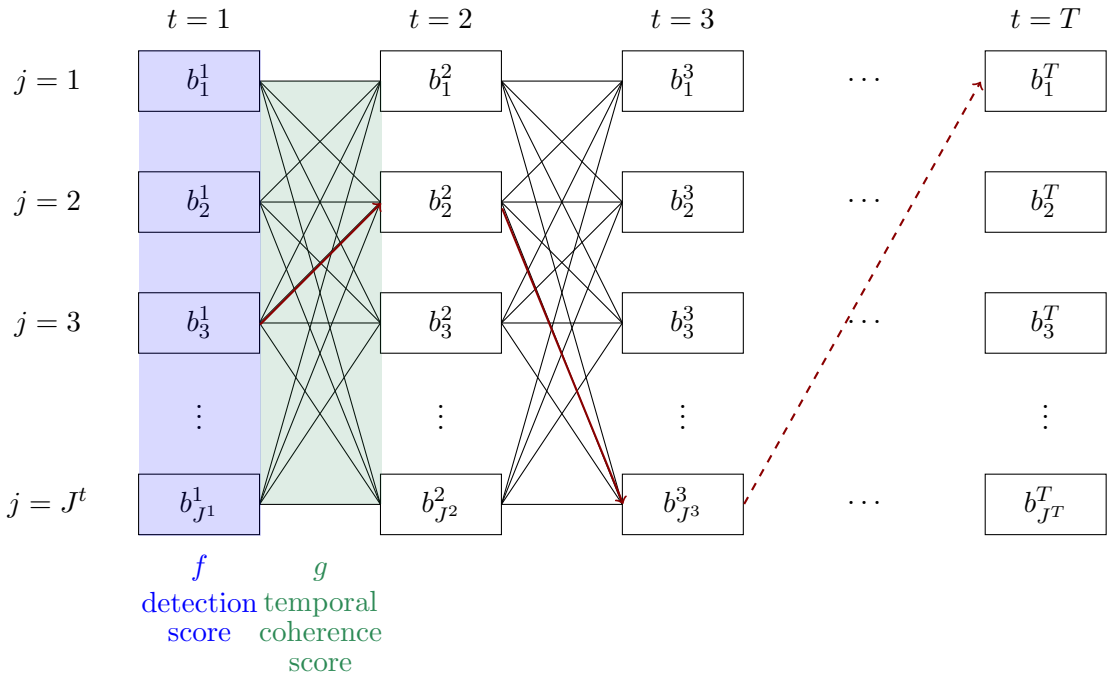


Figure 4: The lattice constructed by the Viterbi algorithm for detection-based tracking. The columns correspond to video frames $t = 1, \ldots, T$. Each column contains the overgenerated collection $b_1^t, \ldots, b_{J^t}^t$ of detections for that frame. The rows correspond to detection indices j. A track $\mathbf{j}$, namely a sequence $j^1, \ldots, j^T$ of detection indices, corresponds to a path through the lattice. The Viterbi algorithm finds the path that optimizes Equation 1, among the exponentially many potential tracks, in time $O(TJ^2)$, where J is the maximum of $J^1, \ldots, J^T$.

This detection-based tracking framework is very general. It can use any detection source(s), any method $f(b)$ for scoring such detections b, and any method $g(b', b)$ for scoring temporal coherence between detections b' and b in adjacent frames. In the work reported here, we use the deformable part model (DPM) detector (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010a; Felzenszwalb, Girshick, & McAllester, 2010b) as the detection source, which yields detections represented as axis-aligned rectangles and use the scores provided by DPM as the basis of $f(b)$. The raw DPM score ranges from $-\infty$ to ∞. Nominally, Equation 1 and the Viterbi algorithm can support such scores. However, these raw DPM scores, unfortunately, are incomparable across object classes. For reasons to be discussed in Section 2.3, joint tracking of multiple objects requires that the detection scores be comparable across their object classes. Moreover, for reasons to be discussed in Section 4, language acquisition requires moderately accurate indication of which object classes are present in the field of view, which could be ascertained if the detection scores were comparable across object classes. To address the above, we normalize the detection scores $f(b)$ within each object class using a sigmoid

$$\frac{1}{1 + \exp(-\chi(f(b) - \rho))}$$

where the parameters χ and ρ are empirically determined per object class so that detection score correlates with the probability of a detection being a true positive. We convert this, and other values discussed in later sections, to log space, to protect against underflow in floating-point calculations. Choosing the parameters χ and ρ in this fashion on a per-class basis allows the resulting detection scores to be comparable across classes. Note that while the resulting values of $f(b)$ are in the range $(-\infty, 0]$, we do not take these to represent log probabilities.

We use optical flow to compute the adjacent-frame temporal-coherence score. We employ the FlowLib optical-flow library (Werlberger, Pock, & Bischof, 2010) as it is one of the highest-performing methods on optical-flow benchmarks (Baker, Scharstein, Lewis, Roth, Black, & Szeliski, 2011). More specifically, to compute $g(b^{t-1}_{j^{t-1}}, b^t_{j^t})$, we compute the optical flow for frame $t-1$, compute the average flow vector v inside the axis-aligned rectangle for detection $b^{t-1}_{j^{t-1}}$, forward project this detection one frame by translating that rectangle along v, and compute the square of the Euclidean distance between the center of that translated rectangle and the center of the corresponding rectangle for $b^t_{j^t}$. This yields a value that measures how well the local detection displacement matches a local estimate of its velocity and ranges from 0 to ∞ in a fashion that is inversely related to temporal coherence. We wish this value to be comparable to the detection score $f(b)$ so that temporal coherence neither overpowers nor is overpowered by detection score. Thus we normalize temporal coherence with a sigmoid as well, using a negative χ to invert the polarity, and convert to log space. Unlike for detection score, a single set of sigmoid parameters can be used across all object classes, because the temporal-coherence score only depends on detection centers. Note that again, while the resulting values of $g(b', b)$ are in the range $(-\infty, 0]$, we do not take these to represent log probabilities. Moreover, even though the values of $f(b)$ and $g(b', b)$ are in the range $(-\infty, 0]$, and the values produced by Equation 1 also lie in that range, they do not represent log probabilities.

2.2 Event Recognition Based on Motion Profile using HMMs

Given a particular track collection, one can determine whether those tracks depict a given event by measuring time-varying properties of those tracks. Such properties could be the relative and/or absolute object positions, velocities, and/or accelerations. The time-varying properties can be represented abstractly as a time-series of feature vectors computed from the tracks. In this view, event recognition can be formulated as time-series classification. Such classification can be performed by hidden Markov models (HMMs), either by computing a likelihood or a MAP estimate. Let us limit consideration, for a moment, to events with a single participant. In this case, we can abstractly take such an HMM to consist of K states, a state-transition function $a(k', k)$ in log space, and an output model $h(k, b)$ which denotes the log probability of generating a detection b in state k. Let us refer to the collection of K, a, and h as an event model λ. In log space, the MAP estimate for a particular track $\mathbf{j}$ is

$$\max_{\mathbf{k}} \left(\sum_{t=1}^{T} h(k^t, b^t_{j^t}) \right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t) \right) \tag{2}$$

where $\mathbf{k}$ is a sequence $k^1, \ldots, k^T$ of states. Let $\mathbf{B_j}$ denote the detection sequence $b^1_{j^1}, \ldots, b^T_{j^T}$ selected from the video clip $\mathbf{B}$ by the track $\mathbf{j}$. Equation 2 constitutes a measure of how

well the detection sequence $\mathbf{B_j}$ selected from a video clip $\mathbf{B}$ by a track $\mathbf{j}$ depicts an event model λ. Higher MAP estimates result from tracks that better depict the event model. MAP estimates can be computed efficiently using the Viterbi algorithm in time $O(TK^2)$. Note the similarity between Equations 2 and 1. This is due to the aforementioned analogy. Momentarily, we will crucially avail ourselves of the fact that both can be computed with the Viterbi algorithm. But we first need to address several subtleties in our formulation.

We use HMMs to encode probability distributions over time-series of feature vectors extracted from object tracks. These in turn serve to represent the meanings of verbs that describe the motion of such participant objects. For example, the meaning of the word *bounce* might be represented with an HMM, like that in Figure 5, that places high probability on a track that exhibits alternating downward and upward motion. While such representations are tolerant of noisy input and can be learned using Baum-Welch (Baum, Petrie, Soules, & Weiss, 1970; Baum, 1972), HMMs with many states, many features, and non-sparsely populated state-transition functions and output models are difficult for humans to understand and create. In Sections 5.3 and 5.4, we conduct experiments with human-generated meaning representations. While, in Section 5.5, we conduct experiments with machine-learned meaning representations, we also compare such with human-generated meaning representations. To facilitate perspicuity in human-generated meaning representations, we adopt a regular-expression notation, such as the following representation of the meaning of the word *bounce*:

$$\lambda_{bounce} \triangleq (\textsc{movingDown}^+ \ \textsc{movingUp}^+)^+$$

In the above, $\textsc{movingDown}(b)$ and $\textsc{movingUp}(b)$ are predicates over detections b that are used to construct the output model $h(k, b)$ and the regular expression is used to determine the number K of states, the state-transition function $a(k', k)$, and which predicate to employ as the output model for a given state. These can be straightforwardly converted to finite-state machines (FSMs) which can, in turn, be viewed as a special case of HMMs with 0/1 state-transition functions and output models ($-\infty/0$ in log space).

Equation 2 is formulated abstractly around a single state-transition function $a(k', k)$. We also must include distributions over initial and final states. Traditional HMM formulations only incorporate initial-state distributions but not final-state distributions. Such HMMs might recognize a prefix of an event specification and not be constrained to match the entire event specification. (Without an initial-state distribution, it might recognize any subinterval of an event specification.) Our actual formulations include such initial- and final-state distributions but we omit them from our presentation for the sake of expository clarity.

Formulating the output model $h(k, b)$ so as to depend on the detections in a single track allows an HMM to encode time-varying constraints on that single track. This can be used to represent the meaning of an intransitive verb that describes the motion of a single participant. We wish, however, to also be able to represent the meanings of transitive verbs that describe the motion of pairs of participants. We accomplish this by extending the output model $h(k, b_1, b_2)$ to depend on pairs of detections, one from each track. If we have two distinct tracks $\mathbf{j}_1 = (j_1^1, \ldots, j_1^T)$ and $\mathbf{j}_2 = (j_2^1, \ldots, j_2^T)$ for two distinct participants, we can think of them as deriving from the same detection pool. This allows extending

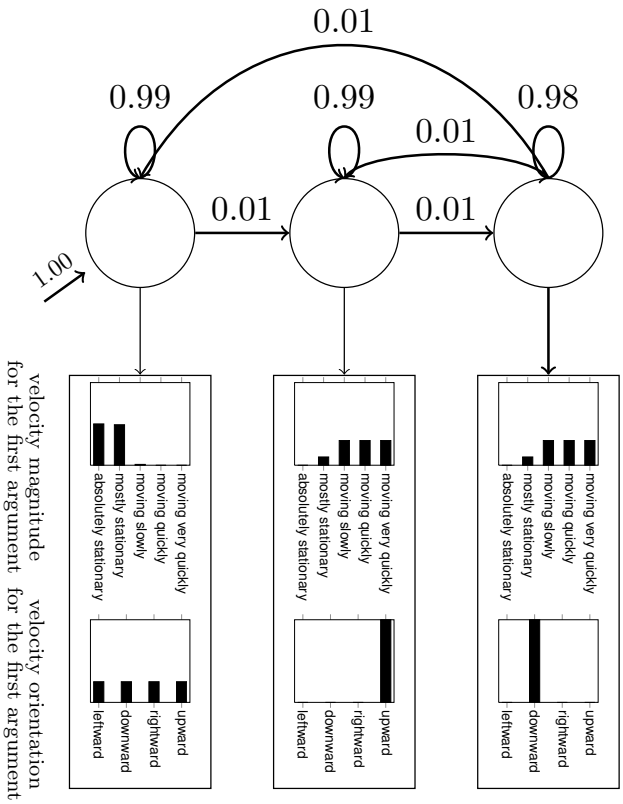


Figure 5: An HMM that represents the meaning of the word *bounce* as a track that exhibits alternating downward and upward motion.

Equation 2 as

$$\max_{\mathbf{k}} \left(\sum_{t=1}^{T} h(k^t, b_{j_1}^t, b_{j_2}^t) \right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t) \right) \tag{3}$$

to support this.

HMMs can be susceptible to short-term noise in the input signal. If one were to have an event model, such as that in Figure 6(a), that is intended to match a time series where there is an interval where the velocity is zero, followed by an interval where there is upward motion, followed by an interval where the velocity is again zero, it may unintentionally match a time series where the interval of upward motion is but a single frame that is spurious and the result of noisy tracking and feature extraction. The same thing might happen with an FSM representation such as

$$\begin{aligned} \textsc{rest}(b_1, b_2) &\triangleq \textsc{stationary}(b_1) \wedge \textsc{stationary}(b_2) \wedge \textsc{close}(b_1, b_2) \\ \textsc{action}(b_1, b_2) &\triangleq \textsc{stationary}(b_1) \wedge \textsc{movingUp}(b_2) \wedge \textsc{close}(b_1, b_2) \\ \lambda_{\textit{pick up}} &\triangleq \textsc{rest}^+ \ \textsc{action}^+ \ \textsc{rest}^+ \end{aligned}$$

that is intended to model the meaning of *pick up* as a period of time where the agent is stationary and close to the patient that is subdivided into three sequential intervals where

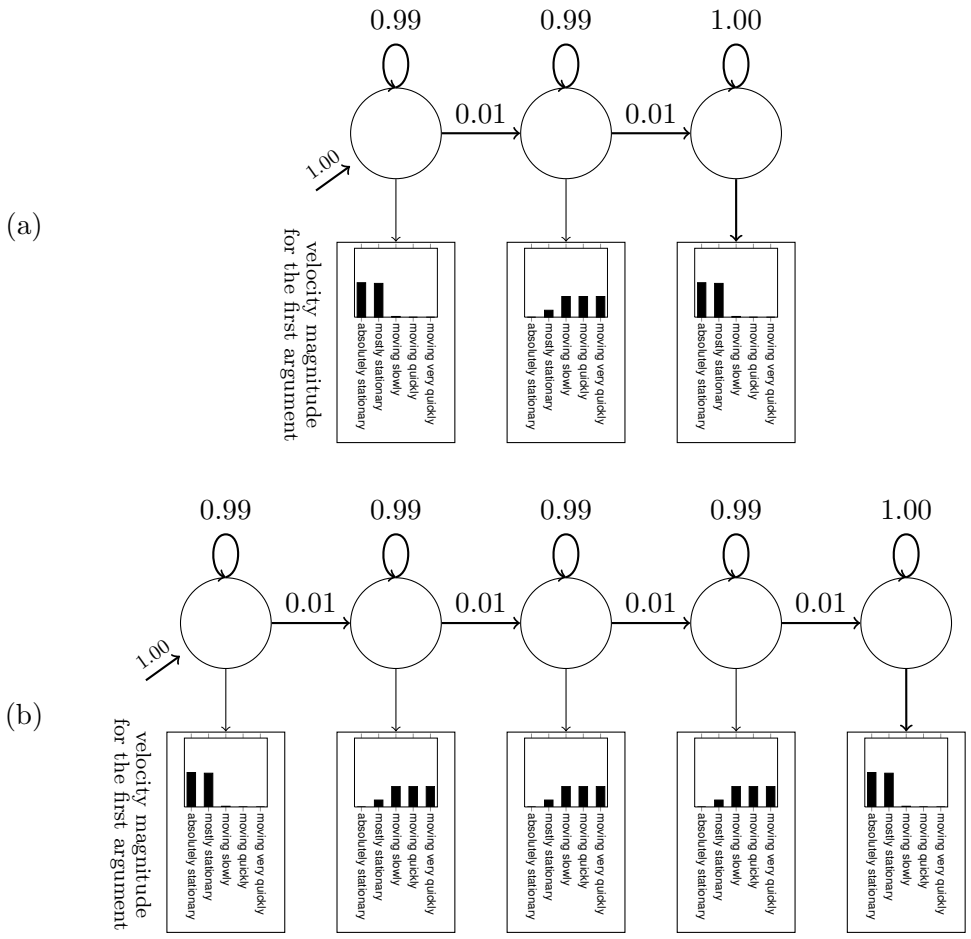


Figure 6: (a) An HMM that is susceptible to short-term noise in the input signal. The central state might admit a noisy impulse lasting a single frame. (b) A variant of (a) that constrains the central interval to hold for at least 3 frames.

the patient is at first stationary, then moves up, and then is stationary again. This can unintentionally match a time series where the patient is continually stationary except for a single frame that is spurious and the result of noisy tracking and feature extraction. We can address this issue by requiring the central interval to have a minimum duration. We indicate such with the regular-expression operator $R^{\{n,\}} \triangleq \underbrace{R \ \ldots \ R}_{n} \ R^*$ to indicate that the R must be repeated at least n times. A definition such as

$$\lambda_{pick\ up} \triangleq \text{REST}^+ \ \text{ACTION}^{\{3,\}} \ \text{REST}^+$$

can be reduced to an FSM within our framework. Similarly, one can add a minimum state-duration requirement to an HMM, such as that in Figure 6(a), by recoding it as in Figure 6(b).

The above handles short-term false positives, namely the presence of a short-term spuriously true signal. We also need to handle short-term false negatives, namely an intended

longer interval where a signal must meet a specified condition but fails to do so due to a short-term failure to meet that condition. We use a new regular-expression operator $R^{[n,]} \triangleq (\text{R } [\text{TRUE}])^{\{n,\}}$ to indicate that R must be repeated at least n times but can optionally have a single frame of noise between each repetition. One can extend HMMs in a similar fashion though we have not found the need to do so because the output models already can tolerate some noise.

Nominally, our detections b_j^t are axis-aligned rectangles represented as image coordinates. This allows the output models $h(k, b)$ to depend on quantities that can be computed from such, e.g., position of the detection center, the size of the detection, and the aspect ratio of the detection, which can indicate notions like *big*, *small*, *tall*, or *wide*. It also allows two-track output models $h(k, b_1, b_2)$ to depend on quantities like the distance between detection centers or the orientation of a line between those centers, which can indicate notions like *close*, *far*, *above*, or *below*. Without further information, it is not possible for the output models to depend on relative or absolute velocity, which would be needed to encode notions like *fast*, *slow*, *stationary*, *moving*, *upwards*, *downwards*, *towards*, or *away from*. One way to achieve such would be to extend the output models to depend on detections from adjacent frames, as in $h(k, b', b)$ or $h(k, b'_1, b_1, b'_2, b_2)$. We can accomplish such with a variant of Equation 2 that sums over pairs of adjacent detections.

$$\max_{\mathbf{k}} \left(\sum_{t=2}^{T} h(k^t, b_{j^{t-1}}^{t-1}, b_{j^t}^t) + a(k^{t-1}, k^t) \right)$$

This can be further generalized by extending the sums over three adjacent frames for acceleration, or even over more frames for longer-term velocity and acceleration. However, multiple-point estimates, e.g., two-point velocity estimates or three-point acceleration estimates, suffer from noise due to inaccurate tracking. Moreover, such extensions would not support other desired features that could be extracted from the image, such as color. Thus we instead extend the notion of detection to include any information that might be extracted from the image at the location of the detection, such as average hue or optical flow inside the detection, and retain the initial formulation of output models $h(k, b)$ and $h(k, b_1, b_2)$ that depends on detections in a single frame.

2.3 The Event Tracker

The aforementioned method operates as a feed-forward pipeline. Equation 1 produces tracks for event participants, a time series of feature vectors is extracted from such tracks, and those time series are classified with HMMs to detect verb/event occurrences. This approach, however, can be very brittle. Failure earlier in the pipeline necessarily leads to failure later in the pipeline. This is particularly of concern, since the pipeline starts with object detections and, as we mentioned before, state-of-the-art object detection is unreliable.

Barbu et al. (2012b) presented a novel approach for addressing this brittleness called the *event tracker*. This approach originates from the observation that Equations 1 and 2 share the same structure due to the aforementioned analogy, and thus share an analogous algorithmic framework for performing the optimization through analogous lattices. The feed-forward pipeline essentially cascades these algorithms and lattices, as shown in Figure 7(a). This independently optimizes Equation 1, as a measure of how well a video clip **B**

depicts a track $\mathbf{j}$, and Equation 2, as a measure of how well the detection sequence $\mathbf{B_j}$ selected from a video clip $\mathbf{B}$ by the track $\mathbf{j}$ depicts an event model λ, performing the former before the latter, and constructing the latter optimization problem around the track $\mathbf{j}$ produced by the former. Doing so takes Equation 2 as the sole measure of how well a video clip $\mathbf{B}$ depicts an event model λ. More precisely, it performs the following optimization:

$$\begin{aligned} &\max_{\mathbf{k}} \left(\sum_{t=1}^{T} h(k^t, b^t_{\hat{j}^t}) \right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t) \right) \\ &\qquad \text{where } \hat{\mathbf{j}} = \operatorname*{argmax}_{\mathbf{j}} \left(\sum_{t=1}^{T} f(b^t_{j^t}) \right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}}, b^t_{j^t}) \right) \end{aligned} \tag{4}$$

While this does measure how well the detection sequence $\mathbf{B_j}$ selected from the video clip $\mathbf{B}$ by the track $\mathbf{j}$ depicts an event model λ, it might not measure how well the video clip $\mathbf{B}$ depicts the event model λ because it fails to incorporate into that measure how well the video clip $\mathbf{B}$ depicts the track $\mathbf{j}$. Thus, we might instead take the sum of Equations 1 and 2 as the measure of how well a video clip $\mathbf{B}$ depicts an event model λ. More precisely, we could adopt the following measure which involves the same optimization as Equation 4:

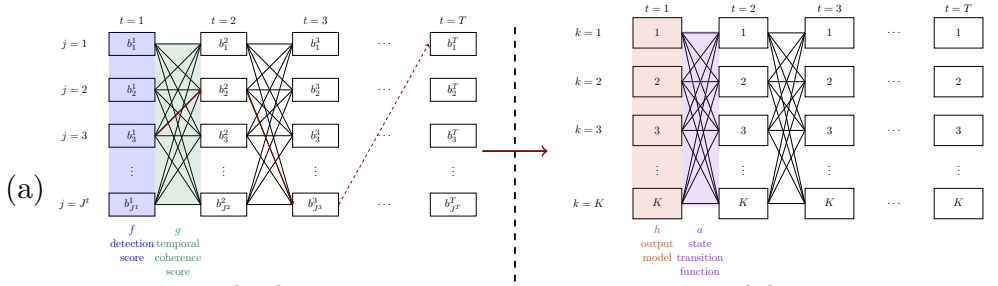
$$\begin{aligned} &\left[\max_{\mathbf{j}} \left(\sum_{t=1}^{T} f(b^t_{j^t}) \right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}}, b^t_{j^t}) \right) \right] + \left[\max_{\mathbf{k}} \left(\sum_{t=1}^{T} h(k^t, b^t_{\hat{j}^t}) \right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t) \right) \right] \\ &\qquad \text{where } \hat{\mathbf{j}} = \operatorname*{argmax}_{\mathbf{j}} \left(\sum_{t=1}^{T} f(b^t_{j^t}) \right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}}, b^t_{j^t}) \right) \end{aligned} \tag{5}$$

This still independently optimizes the track $\mathbf{j}$ with Equation 1 and the state sequence $\mathbf{k}$ with Equation 2. We could, however, attempt to jointly optimize the track $\mathbf{j}$ and the state sequence $\mathbf{k}$. This could be done by lifting both the maximizations over the track $\mathbf{j}$ and the state sequence $\mathbf{k}$ outside the summation of the measures of how well the video clip $\mathbf{B}$ depicts the track $\mathbf{j}$ and how well the detection sequence $\mathbf{B_j}$ selected from the video clip $\mathbf{B}$ by the track $\mathbf{j}$ depicts the event model λ. This leads to the following optimization problem:

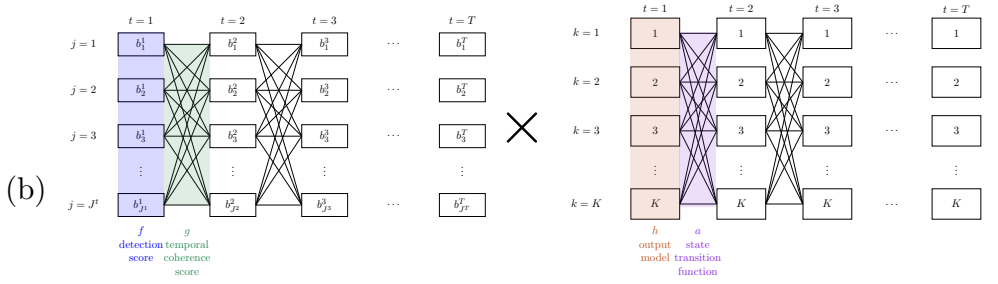
$$\max_{\mathbf{j},\mathbf{k}} \left(\sum_{t=1}^{T} f(b^t_{j^t}) \right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}}, b^t_{j^t}) \right) + \left(\sum_{t=1}^{T} h(k^t, b^t_{j^t}) \right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t) \right) \tag{6}$$

The crucial observation by Barbu et al. (2012b) is that Equation 6 has the same structure as both Equations 1 and 2 and can be optimized using the same Viterbi algorithm by forming a cross-product of the tracker and HMM lattices, as shown in Figure 7(b), where each node in the resulting lattice combines a detection and an HMM state, as shown in Figure 7(c). Since the width of the cross-product lattice is $O(JK)$, applying the Viterbi algorithm to this cross-product lattice finds the path that optimizes Equation 6, among the exponentially many potential paths, in time $O(T(JK)^2)$.

Like before, Equation 6 can trivially be modified to denote a MAP estimate in log space with suitable normalization. However, we do not need to do so because the constant factor introduced by such normalization would not change the result of the joint optimization of the track $\mathbf{j}$ and the state sequence $\mathbf{k}$. Like before, the Viterbi algorithm can be applied to



$$\max_{\mathbf{j}} \left(\sum_{t=1}^{T} f(b_{j^t}^t) \right) + \left(\sum_{t=2}^{T} g(b_{j^{t-1}}^{t-1}, b_{j^t}^t) \right) \qquad \max_{\mathbf{k}} \left(\sum_{t=1}^{T} h(k^t, b_{j^t}^t) \right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t) \right)$$



$$\max_{\mathbf{j},\mathbf{k}} \left(\sum_{t=1}^{T} f(b_{j^t}^t) \right) + \left(\sum_{t=2}^{T} g(b_{j^{t-1}}^{t-1}, b_{j^t}^t) \right) + \left(\sum_{t=1}^{T} h(k^t, b_{j^t}^t) \right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t) \right)$$

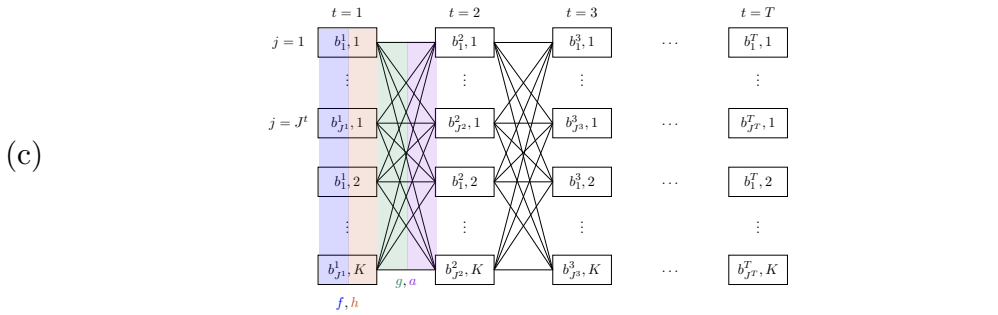


Figure 7: (a) A pipeline consisting of a cascade of a tracker lattice followed by an HMM lattice used for verb/event recognition. In (a), finding the track $\mathbf{j}$ that optimizes the measure of how well a video clip $\mathbf{B}$ depicts that track, Equation 1, happens independently of and prior to finding the state sequence $\mathbf{k}$ that optimizes the measure of how well the detection sequence $\mathbf{B_j}$ selected from a video clip $\mathbf{B}$ by the track $\mathbf{j}$ depicts the event model λ, Equation 2, the latter depending on the track $\mathbf{j}$ produced by the former. Since only the portion from Equation 2 is used as the measure of how well video clip $\mathbf{B}$ depicts event model λ, this corresponds to optimizing the scoring function in Equation 4. Taking the measure of how well a video clip $\mathbf{B}$ depicts an event model λ as a combination of measures of how well the video clip $\mathbf{B}$ depicts the track $\mathbf{j}$ and how well the detection sequence $\mathbf{B_j}$ selected from the video clip $\mathbf{B}$ by the track $\mathbf{j}$ depicts an event model λ can be viewed as optimizing the scoring function in Equation 5, the sum of the two measures. (b) A variant of (a) that jointly optimizes the two measures corresponding to the optimization in Equation 6 that migrates the optimization outside the sum. (c) A method for performing the joint optimization in (b) by forming a cross-product lattice.

Equation 6 without any constraint on permissible values for the detection score $f(b)$, the temporal-coherence score $g(b', b)$, the output model $h(k, b)$, and the state-transition function $a(k', k)$ However, constraining them to lie in the same range empirically allows it to serve as a good scoring function.

The event tracker ameliorates the brittleness of the feed-forward pipeline by allowing top-down information about the event to influence tracking. Using HMMs as event recognizers is accomplished by selecting that event model which best fits the event. This involves running each event model independently on the data. In the context of running a particular event model on the data, that event model could influence tracking in a top-down fashion. For example, in the context of evaluating how well an event model for *walk* fits the data, the tracker would be biased to produce tracks which move at a normal walking pace. Stationary tracks, or those that move too quickly, would not depict the target event and would be filtered out by Equation 6 but not by Equations 1, 4, or 5, when such tracks comprised high-scoring detections and were temporally coherent.

Equation 6 jointly optimizes a single tracker and a single event model. As such, it can only recognize events that have a single participant, such as those described by intransitive verbs. Events with two participants, such as those described by transitive verbs, can be encoded using the methods from Section 2.2, by using Equation 3 instead of Equation 2 and forming the cross product of this with two trackers instead of one.

$$\max_{\mathbf{j}_1,\mathbf{j}_2,\mathbf{k}} \left(\sum_{t=1}^{T} f(b^t_{j_1^t})\right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j_1^{t-1}}, b^t_{j_1^t})\right) + \left(\sum_{t=1}^{T} f(b^t_{j_2^t})\right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j_2^{t-1}}, b^t_{j_2^t})\right) + \left(\sum_{t=1}^{T} h(k^t, b^t_{j_1^t}, b^t_{j_2^t})\right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t)\right) \tag{7}$$

This can be further generalized from two participants to L participants.

$$\max_{\mathbf{J},\mathbf{k}} \left[\sum_{l=1}^{L} \left(\sum_{t=1}^{T} f(b^t_{j_l^t})\right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j_l^{t-1}}, b^t_{j_l^t})\right)\right] + \left(\sum_{t=1}^{T} h(k^t, b^t_{j_1^t}, \ldots, b^t_{j_L^t})\right) + \left(\sum_{t=2}^{T} a(k^{t-1}, k^t)\right) \tag{8}$$

In the above, $\mathbf{J}$ denotes a track collection $\mathbf{j}_1, \ldots, \mathbf{j}_L$ which, in turn, comprises detection indices j_l^t. Equations 7 and 8 can also be optimized with the Viterbi algorithm by forming a cross-product lattice. Since the width of this cross-product lattice is $O(J^L K)$, applying the Viterbi algorithm to this cross-product lattice finds the path that optimizes Equation 8, among the exponentially many potential paths, in time $O(T(J^L K)^2)$. Note that this is exponential in the number L of participants. In practice, however, the arity of the semantic predicate underlying most events is limited, such as to three in the case of ditransitive verbs.

Let $\mathbf{B_J}$ denote the detection-sequence collection $b^1_{j_1^1}, \ldots, b^T_{j_1^T}, \ldots, b^1_{j_L^1}, \ldots, b^T_{j_L^T}$ selected from the video clip $\mathbf{B}$ by the track collection $\mathbf{J}$. Equation 8 jointly optimizes a measure of how well the video clip $\mathbf{B}$ depicts the event model λ as a combination of measures of how well the video clip $\mathbf{B}$ depicts the track collection $\mathbf{J}$ and how well the detection-sequence collection $\mathbf{B_J}$ selected from the video clip $\mathbf{B}$ by the track collection $\mathbf{J}$ depicts an

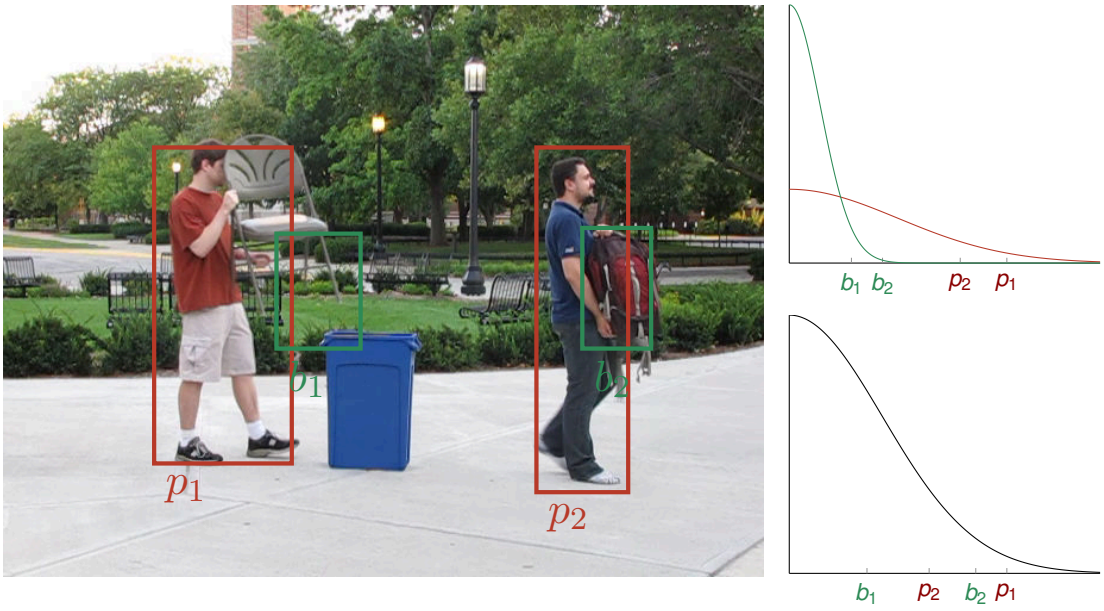


Figure 8: Example showing the necessity for normalization of detection scores across different object classes. (left) Image depicting two pairs of detections for the *person* and *backpack* object classes. (right top) Distribution of raw detection scores for the two object classes. Indicated are scores corresponding to the detections in the image where $f(b_1) = 4$, $f(b_2) = 6$, $f(p_2) = 11$, and $f(p_1) = 14$. (right bottom) Distribution of detection scores for the two object classes after cross-object-class normalization where $f(b_1) = 5$, $f(b_2) = 12$, $f(p_2) = 9$, and $f(p_1) = 14$.

event model λ. Note that Equation 8 involves the summation over multiple detection-score components f, one for each of the L participants. The fact that raw detection scores are incomparable across object class means that the detection scores for different participants contribute to different extents in the final score. Figure 8 shows an example where differences in variance between detection scores for *person* and *trash can* result in a better score through Equation 8 for a spurious set of detections. The values p_1 and p_2 indicate detections for the *person* object class and the values b_1 and b_2 indicate detections for the *backpack* object class. Let us assume that both pairs of detections, (p_1, b_1) and (p_2, b_2), match an event model, such as *carry*, equally well. In that case, the raw detections scores would yield (p_1, b_1) as the best match because $f(p_1) + f(b_1) > f(p_2) + f(b_2)$. It is for this reason that we employ normalization of detection scores as discussed in Section 2.1. Doing so results in the selection of the correct pair of detections, (p_2, b_2), since after normalization, $f(p_2) + f(b_2) > f(p_1) + f(b_1)$.

Figure 9 illustrates the power of the event tracker. The objective is to track the person. However, due to the poor performance of the state-of-the-art person detector, it produces strong false-positive detections on the bench in the background. Even when overgenerating detections, as shown in Figure 9(a), and selecting a track that optimizes Equation 1, as shown in Figure 9(b), this tracks the bench in the background for a portion of the video clip,

Figure 9: Keyframes from a video clip that demonstrates the advantages of the event tracker. (a) Overgenerated person detections. (b) Detections selected by detection-based tracking in Equation 1. Note that it selects a strong false-positive detection on a bench in the background and is not able to rule out such detections as with the exception of a single large jump, the rest of the track happens to be temporally coherent. (c) Detections selected by the event tracker from top-down information, in the form of a model for the transitive verb *carry*, constraining such detections to fill the role of agent in the event, in the context where a backpack, as patient, is carried by the person but not by the bench.

instead of a person. This happens because the track is largely temporally coherent within segments, and in combination with the strong false-positive detections in the background, overpowers the adverse effect of a single large jump, thus yielding a high score for Equation 1. However, top-down information in the form of an event model for the transitive verb *carry*, linked to two trackers, one for an agent and one for a patient, selects a track for the agent, comprising true-positive person detections, that accurately reflects the role played by the person in the event, as shown in Figure 9(c), where a backpack, as patient, is carried by the person and not by the bench in the background.

2.4 The Sentence Tracker

The event tracker from the previous section, and more generally HMM-based event recognizers, can model events with varying numbers of participants (one, two, and L participants

for the event trackers in Equations 6, 7, 8 and one or two participants for the HMM-based event recognizers in Equations 2 and 3). Nominally, we can think of such events as being described by verbs: one-participant events as intransitive verbs, two-participant events as transitive verbs, and three-participant events as ditransitive verbs. Figures 25 through 28 in Appendix B gives examples of HMMs that represent the meanings of verbs. However, nothing in the framework formally restricts us to doing so. The meanings of words in other parts of speech can often also be represented as HMMs. For example, the meaning of a noun that describes an object class can be represented as a single-state one-participant HMM whose output model serves as a classifier for that object class. Figure 23 in Appendix B gives examples of HMMs that represent the meanings of nouns. Similarly, the meaning of an adjective that describes object characteristics can be represented as a single-state one-participant HMM whose output model serves to select detections that exhibit the desired characteristics reflected by that adjective. For example, the meanings of adjectives like *big* or *tall* could be represented with output models over the areas or aspect ratios of participant detections. Likewise, the meaning of a preposition that describes a spatial relation between two objects can be represented as a single-state two-participant HMM whose output model serves to select the collection of features that encode that relation. For example, the meaning of the preposition *to the left of* could be represented with an output model over the relative x-coordinates of the detections for the participants. Figure 24 in Appendix B gives examples of HMMs that represent the meanings of spatial-relation prepositions. More generally, any static property of either a single participant, or a collection of participants, can be encoded as a single-state HMM.

Multiple-state HMMs can encode the dynamic properties of either a single participant or a collection of participants. Such can reflect the meanings of adverbs and prepositions in addition to verbs. For example, the meaning of an adverb such as *quickly* that describes the changing characteristics of the motion of a single participant could be represented as a three-state HMM describing the transition from no motion, to motion with high velocity, back to no motion. Figure 29 in Appendix B gives examples of HMMs that represent the meanings of adverbs. Similarly, the meaning of a preposition such as *towards* that describes the changing relative motion between a pair of participants could be represented as a three-state HMM describing the transition from the agent being distant from the goal, to a period where the distance between the agent and the goal decreases while the goal is stationary, ending with the agent being close to the goal. Figure 30 in Appendix B gives examples of HMMs that represent the meanings of motion prepositions.

We thus see that the distinction between different parts of speech is primarily syntactic, not semantic, *i.e.,* how word use is reflected by the grammar, not its potential meaning. While there may be some coarse-grained trends, such as the canonical structure realizations (CSRs) proposed by Grimshaw (1979, 1981) and Pinker (1984), where nouns typically describe object class, adjectives typically describe object properties, verbs typically describe event class, adverbs typically describe event properties, and prepositions typically describe spatial relations, this is not universally the case. Some intransitive verbs like *sleep* describe a more static object property, some transitive verbs like *hold* describe a more static spatial relation between pairs of objects, and some nouns like *wedding* describe an event. While it might seem like overkill to represent static classifiers as single-state HMMs, there are several advantages to adopting a single uniform meaning representation in the form of

HMMs. First, the capacity for multiple states affords the ability to encode a resilience to temporal noise. Thus in practice, even static properties might be more robustly encoded with multiple states. Second, adopting a single uniform representation simplifies the overall framework and associated algorithms.

The event tracker from the previous section could influence detection-based tracking with top-down information from an event model. This event model could represent the meaning of an individual word. It could constrain a single track for single-participant words like intransitive verbs (Equation 6), a pair of tracks for two-participant words like transitive verbs (Equation 7), or even a collection of L tracks for L-participant words (Equation 8). Just as it was possible to take cross products of multiple trackers with a *single* event model, one can further extend the framework to take cross products of multiple trackers with *multiple* event models, thereby constraining the track collection to jointly satisfy a collection of event models for the words $s_1, \ldots, s_W$ in a sentence $\mathbf{s}$.

$$\max_{\mathbf{J},\mathbf{K}} \left[\sum_{l=1}^{L} \left(\sum_{t=1}^{T} f(b^t_{j^t_l}) \right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}_l}, b^t_{j^t_l}) \right) \right] + \left[\sum_{w=1}^{W} \left(\sum_{t=1}^{T} h_{s_w}(k^t_w, b^t_{j^t_1}, \ldots, b^t_{j^t_L}) \right) + \left(\sum_{t=2}^{T} a_{s_w}(k^{t-1}_w, k^t_w) \right) \right] \tag{9}$$

In the above, $\mathbf{K}$ denotes a state-sequence collection $\mathbf{k}_1, \ldots, \mathbf{k}_W$ which, in turn, comprises state indices k^t_w. This has L distinct trackers with distinct detection indices j^t_l that select the optimal detection for participant l in frame t.

We distinguish between words in the lexicon and occurrences of those in sentences. We refer to the former as lexical entries e and the latter as words w. A given lexical entry may appear as more than one word in a sentence. A lexicon Λ contains E event models $\lambda_1, \ldots, \lambda_E$, one event model λ_e for each lexical entry e. A sentence $\mathbf{s}$ is formulated as a sequence $s_1, \ldots, s_W$ of W lexical entries s_w, one for each word w. Equation 9 has W distinct event models λ_{s_w}, one for each word w in the sentence $\mathbf{s}$, each taken as the event model for the lexical entry s_w for that word w. Each event model λ_{s_w} has distinct numbers K_{s_w} of states, state-transition functions a_{s_w}, and output models h_{s_w}. Note that while the state-transition functions a_{s_w} and output models h_{s_w} vary by word w, the detection score f and the temporal-coherence score g do not vary by participant l.

As formulated in Equation 9, the output model $h_{s_w}(k^t_w, b^t_{j^t_1}, \ldots, b^t_{j^t_L})$ for each word w depends on the detections for frame t selected by the tracks $\mathbf{j}_1, \ldots, \mathbf{j}_L$ for all L participants. In practice, the meaning of each individual word only applies to a subset of the participants, as illustrated in Figure 10. Here, the sentence *The person to the left of the stool carried the traffic cone towards the trash can* describes an event that has four participants: an agent, a referent, a patient, and a goal. The nouns *person*, *stool*, *traffic cone* and *trash can* refer to the agent, referent, patient, and goal respectively. The verb *carried* describes a semantic relation only between the agent and the patient. The preposition *to the left of* describes a semantic relation only between the agent and the referent. The preposition *towards* describes a semantic relation only between the agent and the goal. We employ a *linking function* θ^i_w to indicate which participant fills argument i for the event model for word w. Let $\mathbf{B}\langle \mathbf{s}, t, w, \mathbf{J} \rangle$ denote $b^t_{j^t_{\theta^1_w}}, \ldots, b^t_{j^t_{\theta^{I_{s_w}}_w}}$, the collection of detections selected in

frame t by the track collection $\mathbf{J}$ as assigned to the I_{s_w} arguments of the event model for word w by the linking function θ. We incorporate the arity I in an event model λ, along with the number K of states, the state-transition function a, and the output model h. This allows reformulating Equation 9 as

$$\max_{\mathbf{J},\mathbf{K}} \left[\sum_{l=1}^{L}\left(\sum_{t=1}^{T} f(b^t_{j^t_l})\right) + \left(\sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}_l}, b^t_{j^t_l})\right)\right] + \left[\sum_{w=1}^{W}\left(\sum_{t=1}^{T} h_{s_w}(k^t_w, \mathbf{B}\langle \mathbf{s}, t, w, \mathbf{J}\rangle)\right) + \left(\sum_{t=2}^{T} a_{s_w}(k^{t-1}_w, k^t_w)\right)\right] \tag{10}$$

We refer to Equation 10 as the *sentence tracker*. For the remainder of this paper, $I_{s_w} \leq 2$.

Equation 10 can also be optimized with the Viterbi algorithm by forming a cross-product lattice. Since the width of this cross-product lattice is $O(J^L K^W)$, where K is the maximum of $K_{s_1}, \ldots, K_{s_W}$, applying the Viterbi algorithm to this cross-product lattice finds the path that optimizes Equation 10, among the exponentially many potential paths, in time $O(T(J^L K^W)^2)$. Note that this is exponential both in the number L of participants and the sentence length W. In practice, however, natural-language sentences have bounded length and are typically short. Moreover, the quadratic time complexity is mitigated somewhat by the fact that K^W is an approximation to $\prod_{w=1}^{W} K_{s_w}$. In practice, nouns, adjectives, and spatial-relation prepositions describe static properties of tracks and thus have word models where $K_{s_w} = 1$. Even longer sentences will be comprised predominantly of such word models and will contain relatively few verbs, adverbs, and motion prepositions.

Modeling the meaning of a sentence through a collection of words whose meanings are modeled by HMMs defines a *factorial* HMM for that sentence, where the overall Markov process for that sentence is factored into independent component processes (Brand, Oliver, & Pentland, 1997; Zhong & Ghosh, 2001) for the individual words. In this view, $\mathbf{K}$ denotes the state sequence for the combined factorial HMM and $\mathbf{k}_w$ denotes the factor of that state sequence for word w. Figure 11 illustrates the formation of the cross product of two tracker lattices (Equation 1) and three word lattices (Equation 2), linked together by an appropriate linking function θ to implement the sentence tracker (Equation 10) for the sentence *The person carried the backpack*. Figure 12 illustrates the resulting cross-product lattice where each node in the lattice consists of the combination of two detections, one for each tracker lattice, and three HMM states, one for each word lattice. The state thus represented by each node in this cross-product lattice can be factored into a collection of states written inside the node separated by commas.

Equation 10 constitutes $\mathcal{S} : (\mathbf{B}, \mathbf{s}, \Lambda) \mapsto (\tau, \mathbf{J})$. It scores a video-sentence pair with a measure of how well a given video clip $\mathbf{B}$ depicts a given sentence $\mathbf{s}$, as interpreted by a given lexicon Λ. Alternatively, that score measures how well a given sentence $\mathbf{s}$, as interpreted by a given lexicon Λ, describes a given video clip $\mathbf{B}$. T and $J^1, \ldots, J^T$ are determined from $\mathbf{B}$, W is determined from $\mathbf{s}$, the arities I_{s_w}, the numbers K_{s_w} of states, the state-transition functions a_{s_w} and the output models h_{s_w} are taken from the words models λ_{s_w}, and the number L of participants and the linking function θ are computed from the sentence $\mathbf{s}$ by the linking process $\Theta : \mathbf{s} \mapsto (L, \theta)$ described in Section 3. The result of Equation 10 constitutes the video-sentence score τ. The track collection that yields that score constitutes $\mathbf{J}$.

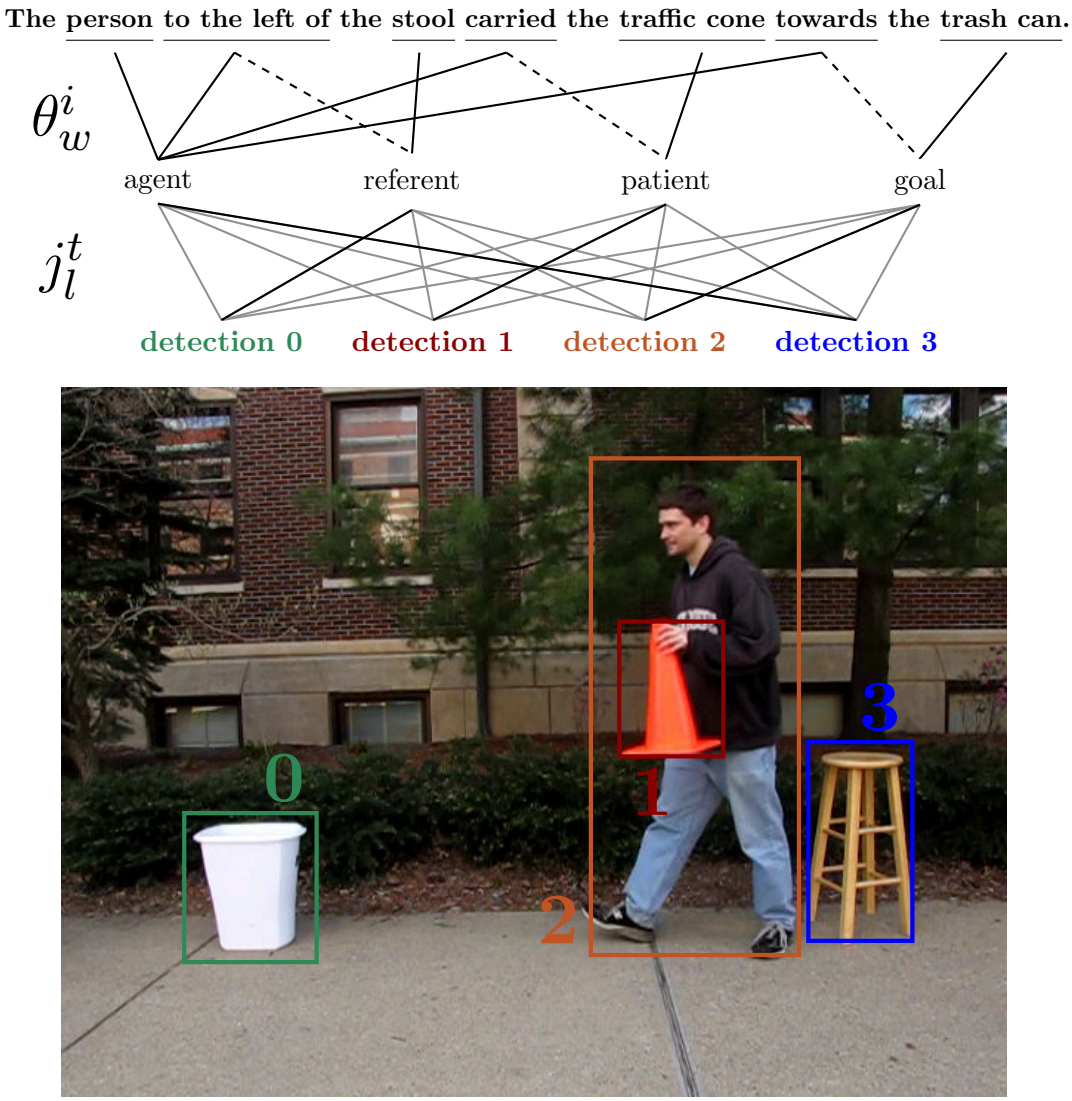


Figure 10: An illustration of the linking function θ used by the sentence tracker. Each word in the sentence has one or more arguments. (When words have two arguments, the first argument is indicated by a solid line and the second by a dashed line.) Each argument of each word is filled by a participant in the event described by the sentence. A given participant can fill arguments for one or more words. Each participant is tracked by a tracker which selects detections from a pool of detections produced by multiple object detectors. The upper mapping θ^i_w from arguments i of words w to participants is determined by parsing the sentence. The lower mapping j^t_l from participants l in frames t to detections is determined automatically by Equation 10. This figure shows a possible (but erroneous) interpretation of the sentence where the lower mapping, indicated by the darker lines, is: agent $\mapsto$ **detection 3**, referent $\mapsto$ **detection 0**, patient $\mapsto$ **detection 1**, and goal $\mapsto$ **detection 2**.

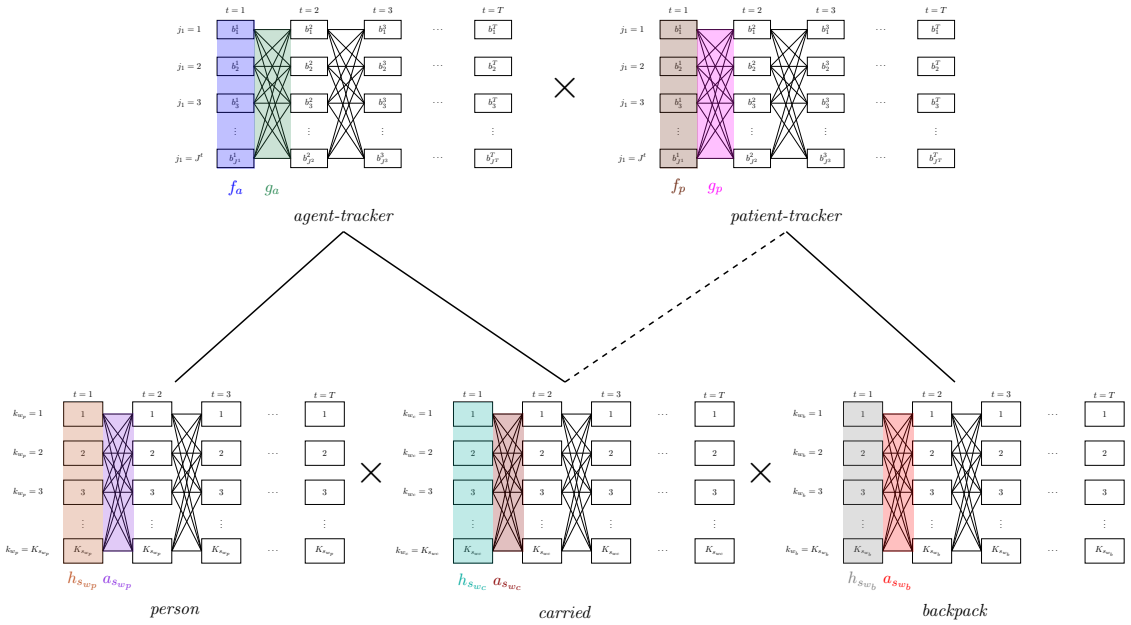


Figure 11: Forming the cross product of two tracker lattices (Equation 1) and three word lattices (Equation 2) to implement the sentence tracker (Equation 10) for the sentence *The person carried the backpack*. The connections between the tracker lattices and the word lattices denote the linking function θ.

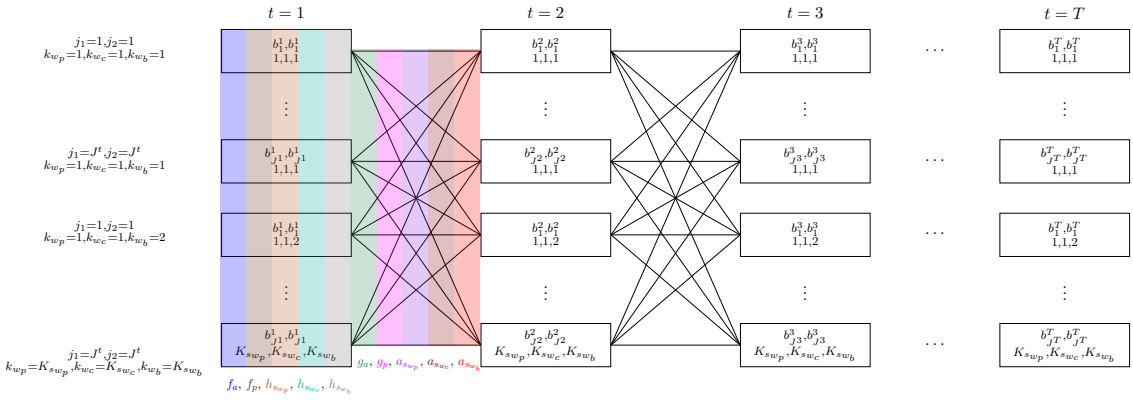


Figure 12: The actual cross-product lattice produced for the example in Figure 11. Note that each node in the lattice consists of the combination of two detections, one for each tracker lattice, and three HMM states, one for each word lattice.

3. The Linking Process

The sentence tracker requires specification of the number L of participants and the linking function θ_w^i that indicates which participant fills argument i of word w for each argument of each word in the sentence. Often, the same participant (*i.e.,* tracker) can fill multiple arguments of multiple words. A sentence like

$$\textit{The } \underbrace{\textit{person}}_{1} \underbrace{\textit{to the right of}}_{2} \textit{ the } \underbrace{\textit{chair}}_{3} \underbrace{\textit{picked up}}_{4} \textit{ the } \underbrace{\textit{backpack}}_{5} \tag{11}$$

has 3 participants and requires a linking function like

$$\theta_1^1 = 1 \quad \theta_2^1 = 1 \quad \theta_2^2 = 2 \quad \theta_3^1 = 2 \quad \theta_4^1 = 1 \quad \theta_4^2 = 3 \quad \theta_5^1 = 3 \tag{12}$$

that assigns the argument of *person* and the first argument of both *to the right of* and *picked up* to the first participant, the argument of *chair* and the second argument of *to the right of* to the second participant, and the argument of *backpack* and the second argument of *picked up* to the third participant. The number L of participants for a sentence $\mathbf{s}$, and the corresponding linking function θ, are produced by a linking process $\Theta : \mathbf{s} \mapsto (L, \theta)$.

We use a particular linking process that is described in details in Appendix A. This process makes use of techniques from mainstream linguistics, namely X-bar theory (Jackendoff, 1977) and government relations (Chomsky, 1982; Aoun & Sportiche, 1983; Haegeman, 1992; Chomsky, 2002). As such, it is limited to a small hand-built grammar (Figure 11a) and a small lexicon (Figure 11b). For our purposes, this is not restrictive. The state of the art in computer vision limits the number of distinct object classes that can be reliably detected and the number of distinct action classes that can be reliably detected. This restricts the number of nouns and verbs that can be supported by any method, such as ours, that attempts to ground language in computer vision methods that detect objects and actions. This further restricts the class of utterances that can be constructed from a small set of nouns and verbs. For this, a small hand-constructed grammar suffices. While one could conceivably use methods that support larger grammars and vocabularies that process a larger space of unrestricted text, it would not be possible to ground such in the current state-of-the art computer vision techniques. We discuss this in further detail in Sections 6 and 7.

The linking process that we employ uses well-known techniques from mainstream linguistics. It is *not* the central contribution of our work. Rather, the central contribution is the sentence tracker (Section 2.4). All the sentence tracker requires is *any* linking process $\Theta : \mathbf{s} \mapsto (L, \theta)$ that maps a sentence $\mathbf{s}$ to the number L of participants and a linking function θ. This need *not* be restricted to the particular grammar and lexicon from Figure 11. Indeed, it can employ *any* one of a plethora of well-known and well-understood techniques that are common in the computational linguistics community. It need not even be restricted to *any* particular grammar or lexicon. It is possible to construct a linking process with standard mechanisms, such as the dependency relations produced by parsing with a dependency grammar. For example, the Stanford Parser (Klein & Manning, 2003) produces the dependencies on the right for the sentence in Equation 11, which can also be used to determine the requisite number of participants and to construct the requisite linking function. The output below correctly identifies three participants, *person-2*, *chair-8*, and

```
det(person-2, The-1)
nsubj(picked-9, person-2)
det(right-5, the-4)
prep_to(person-2, right-5)
det(chair-8, the-7)
prep_of(right-5, chair-8)
root(ROOT-0, picked-9)
prt(picked-9, up-10)
det(backpack-12, the-11)
dobj(picked-9, backpack-12)
```

backpack-12. Note how the transitive verb *picked-9* distinguishes between its two arguments, identifying *person-2* as its first argument through the `nsubj` dependency and *backpack-12* as its second argument through the `dobj` dependency. Also note how the spatial relation *right-5* distinguishes between its two arguments, identifying *person-2* as its first argument through the `prep_to` dependency and *chair-8* as its second argument through the `prep_of` dependency.

4. Language Acquisition with the Sentence Tracker

Children learn language through exposure to rich perceptual context. They observe events while hearing descriptions of such events. By correlating many events with corresponding descriptions, they learn to map words, phrases, and sentences to meaning representations that refer to the world. They come to know that the noun *chair* refers to an object class which typically has a back and four legs. They also come to know that the verb *approach* refers to a dynamic process in which one object moves towards another. These learned concepts are not purely symbolic; they can be used to decide presence or absence of the intended reference in perceptual input. Thus these concepts are *perceptually grounded*.

When children learn language, they are not usually given information about which words in a sentence correspond to which concepts they see. For example, a child who hears *The dog chased a cat* while seeing a dog chase a cat, with no prior knowledge about the meaning of any word in this sentence, might entertain at least two possible correspondences or mappings: (i) $dog \mapsto \textbf{dog} \wedge cat \mapsto \textbf{cat}$ or (ii) $dog \mapsto \textbf{cat} \wedge cat \mapsto \textbf{dog}$. With the first, the child might assume that *chased* means **ran after** while in the second the child might assume that it means **ran before**. Thus a child who hears a description in the context of an observed event will need to disambiguate among several possible interpretations of the meanings of the words in that description. Things get worse when this process exhibits *referential uncertainty* (Siskind, 1996): multiple simultaneous descriptions in the context of multiple simultaneous events.

This situation faced by children motivates the formulation shown in Figure 13, where video clips represent what children see and textual sentences represent what they hear. Note that a given video clip can be paired with more than one sentence and a given sentence can be paired with more than one video clip. Siskind (1996, 2001) showed that even with referential uncertainty and noise, a system based on *cross-situational learning* (Smith, Smith, Blythe, & Vogt, 2006; Smith, Smith, & Blythe, 2011) can robustly acquire a lexicon, mapping words to word-level meanings from sentences paired with sentence-level meanings. However, it did so only for symbolic representations of word- and sentence-level meanings that were not

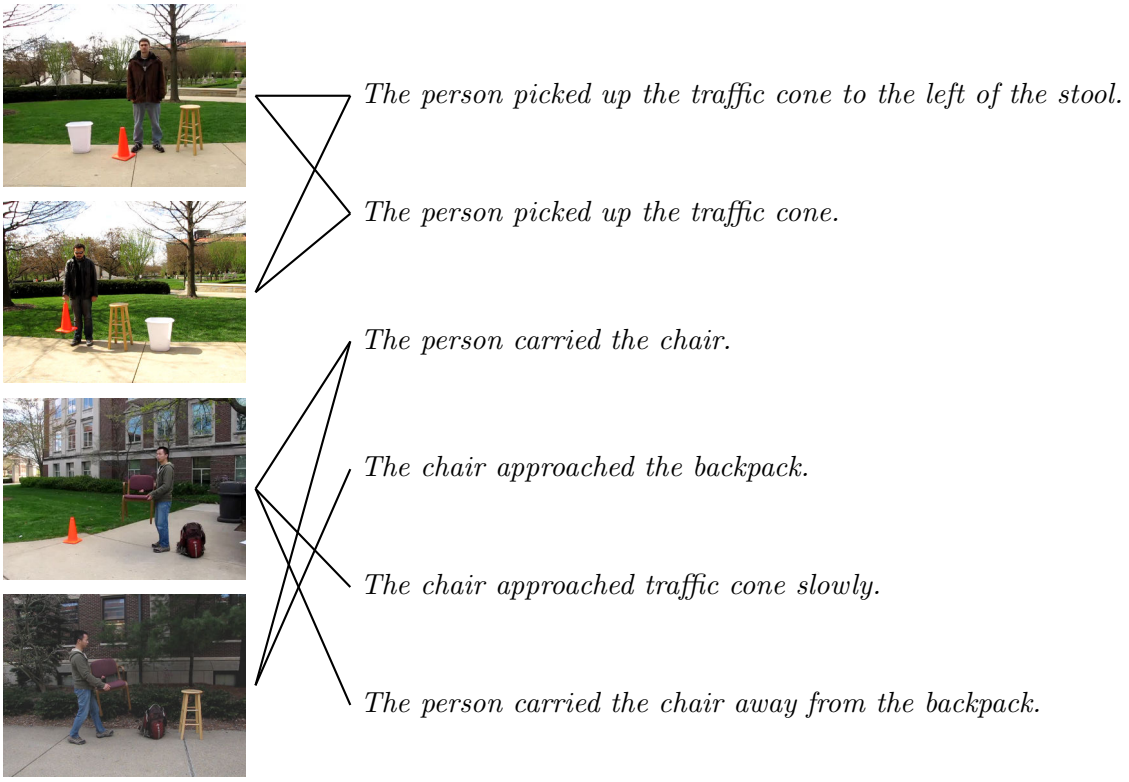


Figure 13: Video-sentence pairs in the language-acquisition problem. A video clip can be paired with multiple sentences and a sentence can be paired with multiple video clips.

perceptually grounded. An ideal system would not require detailed word-level labelings to acquire word meanings from video but rather could learn language in a largely unsupervised fashion, just as a child does, from video paired with sentences. The algorithm presented in this section can resolve the ambiguity inherent with such referential uncertainty to yield a lexicon with the intended meaning for each word. While this algorithm can solve a problem that is reminiscent to that faced by children, we make no psychological or neurophysiological claims.

One can view the language-acquisition task as a constraint-satisfaction problem (CSP), as depicted in Figure 14. Doing so treats words as variables, each with initially unknown meaning. A video-sentence pair can be viewed as a constraint imposed on the words in that sentence: the words in a sentence are mutually constrained by the requirement that the collection of word meanings allow the sentence to describe the video clip. This constraint will be formulated below using a variant of the sentence tracker from Section 2. Since the same word may appear in different sentences, a sufficient number of video-sentence pairs will form a connected network. We can do two types of inference on this network. First, one can perform *inference across different words in the same sentence*. Suppose we know the meanings of all the words in the sentence except for one. In this case, the meaning of the unknown word can be inferred by applying the video-sentence constraint. For example, in Figure 14, if we know the meaning of *backpack* and *person*, the meaning of *picked up* could

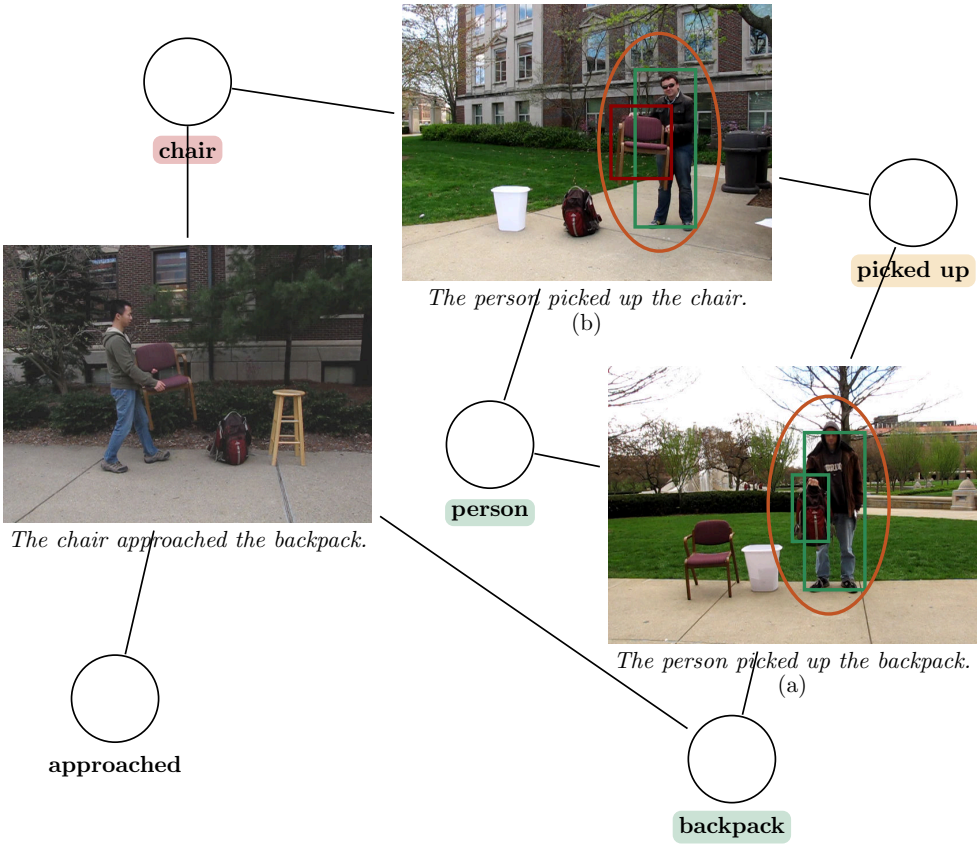


Figure 14: Viewing language acquisition as a constraint-satisfaction problem (CSP) which is solved by propagating information about word meanings around a network. Word meanings in green are used to learn word meanings in orange which are then used to learn further word meanings in red. This performs inference both across different words in the same sentence, and shown in (a), and the same word in different sentences, as shown in (b).

be inferred from constraint (a), because that will be the only process that occurred between the *person* and the *backpack*. Second, one can perform *inference across the same word in different sentences*. The meaning of a given word can be shared and exploited by multiple sentences when inferring the meanings of other words in those sentences. For example, after learning the meaning of *picked up*, from constraint (b), the meaning of *chair* can also be inferred. Thus, information about word meanings can propagate through the network. As a result, word meanings are mutually constrained as they are learned. Siskind (1996) refers to this learning mechanism as cross-situational learning. In practice, this process starts with no information about any word meanings. But our formulation below using EM (Dempster, Laird, & Rubin, 1977) can propagate partial information about word meanings. Thus by starting with an initial guess at the meaning for each word and iterating this process, we can converge to the intended lexicon.

As discussed earlier, the sentence tracker supports representing word meanings as HMMs or as FSMS, a special case of HMMs where the state-transition functions and output mod-

els are 0/1 ($-\infty/0$ in log space). In Section 5.2, we formulate the output models for manually-constructed FSMs as regular expressions over Boolean features computed from the detections using the predicates shown in Table 6. Our procedure for learning word meanings employs HMMs where the state-transition functions and output models are not 0/1. In this case, the output models are derived from the features shown in Table 8. We use Φ to denote the computation that produces the feature vectors from detections and N to denote the length of such feature vectors. Word models λ are extended to incorporate N and Φ.

We employ discrete distributions for our output models h. Further, we assume such distributions are factorial in the features, *i.e.,* the distributions over the features in the feature vector are independent. To this end, we quantize each feature into bins. The particular binning process is described in Section 5.5. This means that the output models take the form

$$h_e(k, b_1, \ldots, b_{I_e}) = \sum_{n=1}^{N} h_e^n(k, \Phi_e^n(b_1, \ldots, b_{I_e}))$$

where

$$\Phi_e^n(b_1, \ldots, b_{I_e}) \in \{\phi_{e,1}^n, \ldots, \phi_{e,Z_e^n}^n\}$$

Z_e^n indicates the number of bins for feature n for lexical entry e and $\phi_{e,z}^n$ indicates the quantized value for bin z of feature n for lexical entry e.

Our learning procedure makes five assumptions.

1. Our training set contains M samples, each pairing a short video clip $\mathbf{B}_m$ with a sentence $\mathbf{s}_m$ that describes that clip. The procedure is not able to determine the alignment between multiple sentences and longer video segments. Note that there is no requirement that the clip depict *only* that sentence. Other objects may be present and other events may occur. In fact, nothing precludes a training set with multiple copies of the same clip, each paired with a different sentence describing a different aspect of that clip. Similarly, nothing precludes a training set with multiple copies of the same sentence, each paired with a different clip that depicts that sentence. Moreover, our procedure potentially can handle a small amount of noise, where a clip is paired with an incorrect sentence that does not describe the clip.
2. We already have (pre-trained) low-level object detectors capable of detecting instances of our target event participants in individual frames of the video. We allow such detections to be unreliable; our method can handle a moderate amount of false positives and false negatives using techniques from Section 2. We do not need to know the mapping from these object-detection classes to nouns; our procedure determines that. In other words, while our detectors locate and classify objects with symbolic labels like chair, these labels are distinct from lexical entries like *chair*. Our procedure learns the mapping from lexical entries to object-class labels. This mapping need not be one-to-one and can be noisy. Learning such a mapping, however, requires that not all object classes be present at all times, as that would not provide the constraint required to learn such mapping—a lexical entry could correspond to any object class. When such is not the case, we additionally need to identify which object classes are present in the video clip. This is made possible by the fact that detection scores have been rendered comparable, using the normalization process described in Section 2.1,

and thus we can use these normalized scores as an indicator of object presence in the video clip.

3. We know the part of speech c_e associated with each lexical entry e. The particular mapping from lexical entry to part of speech used in the experiments in Section 5.5 is given in Table 11(a).
4. The word models λ for all lexical entries of the same part of speech have the same arity I, the same number K of states, the same feature-vector length N, and the same computation Φ that produces the feature vectors, together with the associated binning process for quantizing the features. These values are known and not learned. The particular values for these parameters used in the experiments in Section 5.5 are given in Table 8.
5. We know the linking process Θ and the grammar and lexicon portion needed to determine the number L of participants and the linking function θ for each training sentence. The particular linking process used in the experiments in Section 5.5 is described in Section 3 using the grammar and lexicon portion from Table 11. We do not know the track collection $\mathbf{J}$ chosen for each training sample. This is determined automatically by the methods from Section 2.

The grammar, portions of the lexicon Λ, namely the components I, K, N, and Φ, and the linking process Θ are prespecified and not learned. Only the state-transition functions a and the output models h^n are learned. One can imagine learning some or all of the grammar, some or all of the nonlearned portions of the lexicon, and perhaps even the linking process Θ, such as done by Kwiatkowski, Goldwater, Zettlemoyer, and Steedman (2012). We leave such for future work.

4.1 The General Approach

We are given a grammar, portions of a lexicon Λ, namely the components I, K, N, and Φ, and a linking process Θ. The lexicon contains E word models λ_e for lexical entries e. We are given a training set of M samples, each a video clip $\mathbf{B}_m$ paired with a sentence $\mathbf{s}_m$. Let $\mathcal{B}$ denote $\mathbf{B}_1, \ldots, \mathbf{B}_M$ and $\mathbf{S}$ denote $\mathbf{s}_1, \ldots, \mathbf{s}_M$. We use the grammar, the nonlearned portions of the lexicon Λ, and the linking process Θ to determine the number L of participants and the linking function θ for each training sentence. If we had the state-transition functions a_e and the output models h^n_e for the word models λ_e in the lexicon Λ, we could instantiate the sentence tracker from Equation 10 on each training sample to compute a video-sentence score τ for that sample. A side effect of doing this would be to compute the track collection $\mathbf{J}$ that yielded that video-sentence score. Moreover, we could compute an aggregate score for the entire training set by summing such per-sample scores. However, we don't know the state-transition functions a_e and the output models h^n_e. These constitute the unknown meanings of the words in our training set which we wish to learn. We jointly learn a_e and h^n_e for all lexical entries e by searching for those that maximize the aggregate score.

4.2 The Learning Procedure

We perform that search by Baum-Welch. While Equation 10 constitutes a score that potentially could be maximized, it is easier to adapt a scoring function that is more like a likelihood calculation, than Equation 10, which is more like a MAP estimate, to the EM

framework. Thus we convert Equation 10 from log space to linear space and replace the max with a $\sum$ to redefine our scoring function as follows:

$$\sum_{\mathbf{J},\mathbf{K}} \left[\prod_{l=1}^{L}\left(\prod_{t=1}^{T} f(b_{j_l^t}^t)\right)\left(\prod_{t=2}^{T} g(b_{j_l^{t-1}}^{t-1}, b_{j_l^t}^t)\right)\right] \left[\prod_{w=1}^{W}\left(\prod_{t=1}^{T} h_{s_w}(k_w^t, \mathbf{B}\langle \mathbf{s}, t, w, \mathbf{J}\rangle)\right)\left(\prod_{t=2}^{T} a_{s_w}(k_w^{t-1}, k_w^t)\right)\right] \tag{13}$$

where f, g, h, and a are in linear space. Recall that Equation 6 jointly maximizes the sum of a measure of how well a video clip $\mathbf{B}$ depicts a track $\mathbf{j}$ and a measure of how well the detection sequence $\mathbf{B_j}$ selected from a video clip $\mathbf{B}$ by the track $\mathbf{j}$ depicts an event model λ. Similarly, Equation 10 jointly maximizes the sum of a measure of how well a video clip $\mathbf{B}$ depicts a track collection $\mathbf{J}$ and a measure of how well the detection-sequence collection $\mathbf{B_J}$ selected from a video clip $\mathbf{B}$ by the track collection $\mathbf{J}$ depicts a given sentence $\mathbf{s}$, as interpreted by a given lexicon Λ. One can maximize just the first component of this latter sum.

$$\max_{\mathbf{J}} \left[\sum_{l=1}^{L}\left(\sum_{t=1}^{T} f(b_{j_l^t}^t)\right) + \left(\sum_{t=2}^{T} g(b_{j_l^{t-1}}^{t-1}, b_{j_l^t}^t)\right)\right] \tag{14}$$

This is a variant of Equation 1 for a track collection. One can similarly convert Equation 14 from log space to linear space and replace the max with a $\sum$ to yield:

$$\sum_{\mathbf{J}} \left[\prod_{l=1}^{L}\left(\prod_{t=1}^{T} f(b_{j_l^t}^t)\right)\left(\prod_{t=2}^{T} g(b_{j_l^{t-1}}^{t-1}, b_{j_l^t}^t)\right)\right] \tag{15}$$

By suitable normalization with a constant factor, Equation 15 can be used to obtain the probability of a particular track collection $\mathbf{J}$ relative to a distribution over all possible track collections where the probability of a given track collection was proportional to the summand. Let us denote this probability of a given track collection $\mathbf{J}$ as $P(\mathbf{J}|\mathbf{B})$.

For a given track collection $\mathbf{J}$, one can similarly maximize just the measure of how well the detection-sequence collection $\mathbf{B_J}$ selected from a video clip $\mathbf{B}$ by the track collection $\mathbf{J}$ depicts a sentence $\mathbf{s}$, as interpreted by a given lexicon Λ.

$$\max_{\mathbf{K}} \left[\sum_{w=1}^{W}\left(\sum_{t=1}^{T} h_{s_w}(k_w^t, \mathbf{B}\langle \mathbf{s}, t, w, \mathbf{J}\rangle)\right) + \left(\sum_{t=2}^{T} a_{s_w}(k_w^{t-1}, k_w^t)\right)\right] \tag{16}$$

This is a variant of Equation 2 for a factorial HMM for multiple words. One can similarly convert Equation 16 from log space to linear space and replace the max with a $\sum$ to yield:

$$\sum_{\mathbf{K}} \left[\prod_{w=1}^{W}\left(\prod_{t=1}^{T} h_{s_w}(k_w^t, \mathbf{B}\langle \mathbf{s}, t, w, \mathbf{J}\rangle)\right)\left(\prod_{t=2}^{T} a_{s_w}(k_w^{t-1}, k_w^t)\right)\right] \tag{17}$$

The summand in Equation 17 is the joint probability of a state sequence $\mathbf{K}$ and $\mathbf{B_J}$ depicting a sentence $\mathbf{s}$, as interpreted by a given lexicon Λ: $P(\mathbf{K}, \mathbf{B_J}|\mathbf{s}, \Lambda) = P(\mathbf{B_J}|\mathbf{K}, \mathbf{s}, \Lambda)P(\mathbf{K}|\mathbf{s}, \Lambda)$.

Equation 17 is the (marginal) probability of $\mathbf{B_J}$ depicting a sentence $\mathbf{s}$, as interpreted by a given lexicon Λ: $P(\mathbf{B_J}|\mathbf{s},\Lambda)$. If we divide Equation 13 by Equation 15 we obtain:

$$\mathbf{L}(\mathbf{B};\mathbf{s},\Lambda) = \sum_{\mathbf{J}} P(\mathbf{J}|\mathbf{B}) P(\mathbf{B_J}|\mathbf{s},\Lambda)$$

This is the expected probability of $\mathbf{B_J}$ depicting a sentence $\mathbf{s}$, as interpreted by a given lexicon Λ, over the track collection distribution underlying $P(\mathbf{J}|\mathbf{B})$. Equations 13 and 15 can both be computed efficiently by the forward algorithm (Baum & Petrie, 1966). This allows us to take $\mathbf{L}(\mathbf{B};\mathbf{s},\Lambda)$ as a sample score and adopt

$$\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda) = \prod_{m=1}^{M} \mathbf{L}(\mathbf{B}_m;\mathbf{s}_m,\Lambda)$$

as the training-set score. We seek the a and h in Λ that maximize $\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda)$. Note that both the sample and training-set scores are in $[0,1]$.

We can find a local maximum to this objective function using the same techniques as used by Baum-Welch. The reestimation formulas can be derived with auxiliary functions that are analogous to those used for HMMs (Bilmes, 1998). Let us first define $\mathcal{J} = \mathbf{J}_1,\ldots,\mathbf{J}_M$ and $\mathcal{K} = \mathbf{K}_1,\ldots,\mathbf{K}_M$ to be track collections and state-sequence collections for the entire training set. Then let us define $\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda)$ as the product of the summand of Equation 13 over the training set divided by the product of Equation 15 over the training set. Thus we have:

$$\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda) = \sum_{\mathcal{J},\mathcal{K}} \mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda)$$

We adopt the following auxiliary function:

$$F(\Lambda,\Lambda') = \sum_{\mathcal{J},\mathcal{K}} \mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda') \log \mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda)$$

where Λ' is the current lexicon and Λ is a potential new lexicon. One can show that $F(\Lambda,\Lambda') \geq F(\Lambda',\Lambda')$ implies $\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda) \geq \mathcal{L}(\mathcal{B};\mathbf{S},\Lambda')$.

$$\begin{aligned}
F(\Lambda,\Lambda') - F(\Lambda',\Lambda') &= \mathcal{L}(\mathcal{B};\mathbf{S},\Lambda') \sum_{\mathcal{J},\mathcal{K}} \left[\frac{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda')}{\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda')} \log \frac{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda)}{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda')} \right] \\
&\propto \sum_{\mathcal{J},\mathcal{K}} \left[P(\mathcal{J},\mathcal{K}|\mathcal{B},\mathbf{S},\Lambda') \log \frac{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda)}{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda')} \right] \\
&\leq \log \sum_{\mathcal{J},\mathcal{K}} \left[P(\mathcal{J},\mathcal{K}|\mathcal{B},\mathbf{S},\Lambda') \frac{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda)}{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda')} \right] \\
&= \log \sum_{\mathcal{J},\mathcal{K}} \frac{\mathcal{L}(\mathcal{B},\mathcal{J},\mathcal{K};\mathbf{S},\Lambda)}{\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda')} \\
&= \log \frac{\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda)}{\mathcal{L}(\mathcal{B};\mathbf{S},\Lambda')}
\end{aligned}$$

The second step above holds because the training-set score $\mathcal{L}(\mathcal{B}; \mathbf{S}, \Lambda')$ is nonnegative. The third step holds due to Jensen's (1906) inequality. Thus given the current lexicon Λ', if we find a new lexicon Λ such that $F(\Lambda, \Lambda') \geq F(\Lambda', \Lambda')$, one can iterate this process, increasing the training-set score to a local maximum. This can be done by maximizing $F(\Lambda, \Lambda')$ with respect to Λ. Since $\mathcal{L}(\mathcal{B}, \mathcal{J}, \mathcal{K}; \mathbf{S}, \Lambda)$ is proportional to the product of the summands of Equation 13 over the training set, which is the product of two terms, only the latter of which depends on Λ, the following holds:

$$\begin{aligned} F(\Lambda, \Lambda') &\propto \sum_{\mathcal{J},\mathcal{K}} \frac{\mathcal{L}(\mathcal{B}, \mathcal{J}, \mathcal{K}; \mathbf{S}, \Lambda')}{\mathcal{L}(\mathcal{B}; \mathbf{S}, \Lambda')} \log \mathcal{L}(\mathcal{B}, \mathcal{J}, \mathcal{K}; \mathbf{S}, \Lambda) \\ &\propto \sum_{\mathcal{J},\mathcal{K}} \frac{\mathcal{L}(\mathcal{B}, \mathcal{J}, \mathcal{K}; \mathbf{S}, \Lambda')}{\mathcal{L}(\mathcal{B}; \mathbf{S}, \Lambda')} \sum_{m=1}^{M} \sum_{w=1}^{W_m} \left(\sum_{t=1}^{T_m} \underbrace{\log h_{s_{m,w}}(k_{m,w}^t, \mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m \rangle)}_{\mathbf{h}} \right) \\ &\qquad + \left(\sum_{t=2}^{T_m} \underbrace{\log a_{s_{m,w}}(k_{m,w}^{t-1}, k_{m,w}^t)}_{\mathbf{a}} \right) \end{aligned}$$

where T_m is the number of frames in the video clip $\mathbf{B}_m$ for training sample m , W_m is the number of words in the sentence $\mathbf{s}_m$ for training sample m, $s_{m,w}$ is the lexical entry for word w in the sentence $\mathbf{s}_m$ for training sample m, and $k_{m,w}^t$ is the state k_w^t in the state-sequence collection $\mathbf{K}_m$ for training sample m. In the above, $\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m \rangle$ is extended to denote $b^t_{j^t_{\theta^1_{m,w}}}, \ldots, b^t_{j^t_{\theta^{I_{s_{m,w}}}_{m,w}}}$, the collection of detections selected in frame t of the video clip $\mathbf{B}_m$ by the track collection $\mathbf{J}_m$ as assigned to the $I_{s_{m,w}}$ arguments of the word model for word w in sentence $\mathbf{s}_m$ by the linking function $\theta^i_{m,w}$ produced on $\mathbf{s}_m$ that determines the participant for argument i of word w for sentence $\mathbf{s}_m$. Thus $F(\Lambda, \Lambda')$ comprises two terms, one of which, $\mathbf{H}$, is a weighted sum of terms $\mathbf{h}$ and the other of which, $\mathbf{A}$, is a weighted sum of terms $\mathbf{a}$. One can maximize $F(\Lambda, \Lambda')$ by maximizing $\mathbf{H}$ and $\mathbf{A}$ independently. These lead to reestimation procedures for the output models h and state-transition functions a.

First consider $\mathbf{A}$. Rewrite the term to explicitly sum over lexical entries e and pairs of states k' and k.

$$\begin{aligned} \mathbf{A} &= \sum \frac{\mathcal{L}(\mathcal{B}, k_{m,w}^{t-1} = k', k_{m,w}^t = k; \mathbf{S}, \Lambda')}{\mathcal{L}(\mathcal{B}; \mathbf{S}, \Lambda')} \log a_e(k', k) \\ &= \sum \frac{\mathbf{L}(\mathbf{B}_m, k_{m,w}^{t-1} = k', k_{m,w}^t = k; \mathbf{s}_m, \Lambda') \mathcal{L}(\mathcal{B}_{m' \neq m}; \mathbf{S}_{m' \neq m}, \Lambda')}{\mathbf{L}(\mathbf{B}_m; \mathbf{s}_m, \Lambda') \mathcal{L}(\mathcal{B}_{m' \neq m}; \mathbf{S}_{m' \neq m}, \Lambda')} \log a_e(k', k) \qquad (18) \\ &= \sum \frac{\mathbf{L}(\mathbf{B}_m, k_{m,w}^{t-1} = k', k_{m,w}^t = k; \mathbf{s}_m, \Lambda')}{\mathbf{L}(\mathbf{B}_m; \mathbf{s}_m, \Lambda')} \log a_e(k', k) \end{aligned}$$

where $\sum$ denotes $\sum_{e=1}^{E}\sum_{k'=1}^{K_e}\sum_{k=1}^{K_e}\sum_{\substack{m=1\\ s_{m,w}=e}}^{M}\sum_{w=1}^{W_m}\sum_{t=2}^{T_m}$ and where

$$\begin{aligned}
\mathcal{L}(\mathcal{B}, k_{m,w}^{t-1}=k', k_{m,w}^{t}=k; \mathbf{S}, \Lambda') &= \sum_{\mathcal{J}} \sum_{\substack{\mathcal{K}\\ k_{m,w}^{t-1}=k'\\ k_{m,w}^{t}=k}} \mathcal{L}(\mathcal{B}, \mathcal{J}, \mathcal{K}; \mathbf{S}, \Lambda') \\
\mathbf{L}(\mathbf{B}_m, k_{m,w}^{t-1}=k', k_{m,w}^{t}=k; \mathbf{s}_m, \Lambda') &= \sum_{\mathbf{J}_m} \sum_{\substack{\mathbf{K}_m\\ k_{m,w}^{t-1}=k'\\ k_{m,w}^{t}=k}} \mathbf{L}(\mathbf{B}_m, \mathbf{J}_m, \mathbf{K}_m; \mathbf{s}_m, \Lambda') \\
\mathcal{L}(\mathcal{B}_{m'\neq m}; \mathbf{S}_{m'\neq m}, \Lambda') &= \prod_{\substack{m'=1\\ m'\neq m}}^{M} \mathbf{L}(\mathbf{B}_{m'}; \mathbf{s}_{m'}, \Lambda)
\end{aligned}$$

The second step in Equation 18 holds because of the assumption that the training samples are i.i.d. Taking the derivative of $\mathbf{A}$ with respect to each $a_e(k', k)$, we get the reestimation formula for the state-transition function:

$$a_e(k', k) := \kappa_e(k') \sum_{\substack{m=1\\ s_{m,w}=e}}^{M}\sum_{w=1}^{W_m}\sum_{t=2}^{T_m} \underbrace{\frac{\mathbf{L}(\mathbf{B}_m, k_{m,w}^{t-1}=k', k_{m,w}^{t}=k; \mathbf{s}_m, \Lambda')}{\mathbf{L}(\mathbf{B}_m; \mathbf{s}_m, \Lambda')}}_{\xi(m,w,k',k,t)}$$

The coefficient $\kappa_e(k')$ is chosen to normalize the distribution so that it sums to one.

The reestimation formula for the output model can be derived similarly from $\mathbf{H}$. We make use of the fact that the output model is a factorial model where the factors are discrete distributions. In linear space:

$$h_e(k, b_1, \ldots, b_{I_e}) = \prod_{n=1}^{N_e} h_e^n(k, \Phi_e^n(b_1, \ldots, b_{I_e}))$$

Again, rewrite $\mathbf{H}$ to explicitly sum over lexical entries e, states k, features n, and bins z.

$$\begin{aligned}
\mathbf{H} &= \sum \frac{\mathcal{L}(\mathcal{B}, k_{m,w}^{t}=k, \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m\rangle)=\phi_{e,z}^n; \mathbf{S}, \Lambda')}{\mathcal{L}(\mathcal{B}; \mathbf{S}, \Lambda')} \log h_e^n(k, \phi_{e,z}^n) \\
&= \sum \frac{\mathbf{L}(\mathbf{B}_m, k_{m,w}^{t}=k, \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m\rangle)=\phi_{e,z}^n; \mathbf{s}_m, \Lambda')\mathcal{L}(\mathcal{B}_{m'\neq m}; \mathbf{S}_{m'\neq m}, \Lambda')}{\mathbf{L}(\mathbf{B}_m; \mathbf{s}_m, \Lambda')\mathcal{L}(\mathcal{B}_{m'\neq m}; \mathbf{S}_{m'\neq m}, \Lambda')} \log h_e^n(k, \phi_{e,z}^n) \\
&= \sum \frac{\mathbf{L}(\mathbf{B}_m, k_{m,w}^{t}=k, \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m\rangle)=\phi_{e,z}^n; \mathbf{s}_m, \Lambda')}{\mathbf{L}(\mathbf{B}_m; \mathbf{s}_m, \Lambda')} \log h_e^n(k, \phi_{e,z}^n)
\end{aligned}$$

where $\sum$ denotes $\sum_{e=1}^{E}\sum_{k=1}^{K_e}\sum_{n=1}^{N_e}\sum_{z=1}^{Z_e^n}\sum_{\substack{m=1\\ s_{m,w}=e}}^{M}\sum_{w=1}^{W_m}\sum_{t=1}^{T_m}$ and where

$$\mathcal{L}(\mathcal{B}, k_{m,w}^t = k, \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m\rangle) = \phi_{e,z}^n; \mathbf{S}, \Lambda') = \sum_{\substack{\mathcal{J}\\ \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m,t,w,\mathbf{J}_m\rangle)=\phi_{e,z}^n}} \sum_{\substack{\mathcal{K}\\ k_{m,w}^t=k}} \mathbf{L}(\mathcal{B}, \mathcal{J}, \mathcal{K}; \mathbf{S}, \Lambda')$$

$$\mathbf{L}(\mathbf{B}_m, k_{m,w}^t = k, \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m\rangle) = \phi_{e,z}^n; \mathbf{s}_m, \Lambda') = \sum_{\substack{\mathbf{J}_m\\ \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m,t,w,\mathbf{J}_m\rangle)=\phi_{e,z}^n}} \sum_{\substack{\mathbf{K}_m\\ k_{m,w}^t=k}} \mathbf{L}(\mathbf{B}_m, \mathbf{J}_m, \mathbf{K}_m; \mathbf{s}_m, \Lambda')$$

Taking the derivative of $\mathbf{H}$ with respect to each $h_e^n(k, \phi_{e,z}^n)$, we get the reestimation formula for the output model:

$$h_e^n(k,\phi) := \psi_e^n(k) \sum_{\substack{m=1\\ s_{m,w}=e}}^{M}\sum_{w=1}^{W_m}\sum_{t=1}^{T_m} \underbrace{\frac{\mathbf{L}(\mathbf{B}_m, k_{m,w}^t = k, \Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m\rangle) = \phi; \mathbf{s}_m, \Lambda')}{\mathbf{L}(\mathbf{B}_m; \mathbf{s}_m, \Lambda')}}_{\delta(m,w,n,k,\phi,t)}$$

The coefficient $\psi_e^n(k)$ is chosen to normalize the distribution so that it sums to one.

The reestimation formulas involve *occurrence counting*. Since we use factorial HMMs that involve a cross-product lattice and use a scoring function derived from Equation 13 that incorporates both tracking (Equation 1) and word models (Equation 2), we need to count occurrences in the whole cross-product lattice. As an example of such cross-product occurrence counting, when counting the transitions from state k' to k for word w from frame $t-1$ to t in sample m, *i.e.,* $\xi(m, w, k', k, t)$, we need to count all the possible paths through the adjacent factorial states, *i.e.,* from $j_{m,1}^{t-1}, \ldots, j_{m,L}^{t-1}, k_{m,1}^{t-1}, \ldots, k_{m,W}^{t-1}$ to $j_{m,1}^{t}, \ldots, j_{m,L}^{t}, k_{m,1}^{t}, \ldots, k_{m,W}^{t}$ such that $k_{m,w}^{t-1} = k'$ and $k_{m,w}^t = k$. Similarly, when counting the frequency of being at state k while observing the value ϕ as the feature n in frame t of sample m for the word w, *i.e.,* $\delta(m, w, n, k, \phi, t)$, we need to count all the possible paths through the factorial state $j_{m,1}^{t}, \ldots, j_{m,L}^{t}, k_{m,1}^{t}, \ldots, k_{m,W}^{t}$ such that $k_{m,w}^t = k$ and $\Phi_e^n(\mathbf{B}_m\langle \mathbf{s}_m, t, w, \mathbf{J}_m\rangle) = \phi$.

The reestimation of one word model can depend on the previous estimate for other word models. This dependence happens because the linking function can assign the same participant to arguments of different words in a sentence and the same lexical entry can appear in different training sentences. *It is precisely this dependence that leads to cross-situational learning: the former performs inference across different words in the same sentence and the latter performs inference across the same word in different sentences.*

5. Experiments

The sentence tracker implements a function $\mathcal{S} : (\mathbf{B}, \mathbf{s}, \Lambda) \mapsto (\tau, \mathbf{J})$ that takes a video clip $\mathbf{B}$ as input, along with a sentence $\mathbf{s}$ and a lexicon Λ, and produces, as output, a video-sentence score τ, together with a track collection $\mathbf{J}$ that depicts the sentence $\mathbf{s}$ as interpreted by the lexicon Λ. The ability to produce both a score and a track collection allows the sentence tracker to be used in a variety of ways, among them:

language inference Using the track collection that it produces, it can take a sentence as input and focus its attention on the event described in the sentence. This allows processing a video clip that depicts many participants, various subsets of whom are engaged is different events, to track those particular participants that are engaged in a particular event as specified by a sentence.

language generation Using the score that it produces, it can generate sentential descriptions of video clips by efficiently searching through the space of possible sentences to find one that best describes a given clip.

language acquisition Using the score that it produces, it can learn word meanings from a training set of video clips paired with sentences that describe those clips, by searching the space of potential word meanings to find those that collectively allows the sentences to best describe the associated clips.

We evaluate the first use in Section 5.3, the second use in Section 5.4, and the third use in Section 5.5.

5.1 The Corpora

To conduct our evaluation, we filmed two different corpora, each containing 94 video clips. One corpus was used for the experiments in Sections 5.3 and 5.4 while the other was used for the experiments in Section 5.5. Both corpora were filmed at 640×480 resolution and 30 fps. Each contained clips that varied in length between 3 and 5 seconds. Both were filmed in a variety of outdoor environments, the first varying between three different environments and the second varying between four. The camera was moved between filming each clip so that the varying background precluded unanticipated confounds.

The video clips were filmed with a variety of actors and objects. The clips in the first corpus each contain one or two people from a collection of three actors while the clips in the second corpus each contain a single person from a collection of four actors. The first corpus was filmed with three objects, a backpack, a chair, and a trash can, each of which were present in the field of view for all clips. The second corpus was filmed with five objects, a backpack, a chair, a traffic cone, a trash can, and a stool, with either two or three present in the field of view of any given clip. The whole dataset was counterbalanced to avoid artifactual correlation. Each object class and combination of object classes appears in clips with nearly equal frequency.

The four different environments for the second corpus were used to construct three different cross-validation folds. The 29 video clips filmed in one environment always contain exactly two objects while the 23, 22, and 20 clips filmed in each of the other three environments respectively always contain exactly three objects. The test set for a given fold comprised all of the clips filmed in one of the latter three environments. Thus the test sets for the three folds contained 23, 22, and 20 clips respectively. The training set for a given fold comprised all clips except for the test set for that fold. Thus the training sets for the three folds contained 71, 72, and 74 clips respectively.

All video clips depict multiple simultaneous events. The depiction, from clip to clip, varied in scene layout and the actor(s) performing the event. The clips in the first corpus each depicted one or more of the 21 sentences from Table 1. The clips in the second corpus each depicted one or more of the 187 sentences from Tables 2 and 3. These sentences

1 a. *The backpack approached the trash can.*
b. *The chair approached the trash can.*
2 a. *The red object approached the trash can.*
b. *The blue object approached the trash can.*
3 a. *The person to the left of the trash can put down an object.*
b. *The person to the right of the trash can put down an object.*
4 a. *The person put down the trash can.*
b. *The person put down the backpack.*
5 a. *The person carried the red object.*
b. *The person carried the blue object.*
6 a. *The person picked up an object to the left of the trash can.*
b. *The person picked up an object to the right of the trash can.*
7 a. *The person picked up an object.*
b. *The person put down an object.*
8 a. *The person picked up an object quickly.*
b. *The person picked up an object slowly.*
9 a. *The person carried an object towards the trash can.*
b. *The person carried an object away from the trash can.*
10. *The backpack approached the chair.*
11. *The red object approached the chair.*
12. *The person put down the chair.*

Table 1: A selection of sentences drawn from the grammar in Table 11(a) based on which we collected multiple video clips for the first corpus. Note that sentence pairs 1 through 9 constitute minimal pairs, where a single constituent varies between two lexical entries in each pair. The varying constituent ranges over all parts of speech and all sentential positions.

were constrained to conform to the grammar in Table 11(a). The 187 sentences for the second corpus were divided into two groups, one consisting of 175 sentences that were used exclusively for training and one consisting of 12 sentences that were used exclusively for test. This delineation is indicated by the horizontal line in Table 3.

The corpora were carefully constructed in a number of ways. First, many video clips depict more than one sentence. In particular, many clips depict simultaneous distinct events. Second, each sentence describes multiple clips. Third, the first corpus was constructed with minimal pairs: clips described by a pair of sentences which differ in exactly one lexical item. These minimal pairs help evaluate language inference and are indicated as the 'a' and 'b' variants of sentences 1–9 in Table 1. That varying lexical item was carefully chosen to span all parts of speech and all sentential positions: sentence 1 varies subject noun, sentence 2 varies subject adjective, sentence 3 varies subject preposition, sentence 4 varies object noun, sentence 5 varies object adjective, sentence 6 varies object preposition, sentence 7 varies verb, sentence 8 varies adverb, and sentence 9 varies motion preposition. Fourth, each clip in the second corpus contains only a subset of the objects used in that corpus. Without such asymmetry it would be difficult (but not impossible) to determine the correspondence

The chair approached the stool.
The chair to the right of the backpack approached the stool.
The chair to the left of the stool approached the stool.
The person picked up the stool.
The person picked up the stool to the left of the backpack.
The person carried the trash can.
The person carried the trash can to the left of the backpack.
The person put down the trash can.
The person put down the trash can quickly.
The person put down the trash can to the left of the stool.
The person to the left of the backpack put down the trash can.
The person picked up the chair.
The person picked up the chair quickly.
The person picked up the chair to the left of the traffic cone.
The person picked up the chair to the left of the backpack.
The person put down the chair.
The person put down the chair quickly.
The person to the left of the traffic cone put down the chair.
The person carried the traffic cone.
The person to the left of the backpack carried the traffic cone.
The person carried the traffic cone away from the trash can.
The backpack approached the traffic cone.
The backpack to the right of the chair approached the traffic cone.
The backpack to the left of the traffic cone approached the traffic cone.
The person put down the traffic cone.
The person put down the traffic cone to the left of the stool.
The person to the left of the chair put down the traffic cone.
The person carried the backpack.
The person to the left of the chair carried the backpack.
The person carried the backpack away from the stool.
The person put down the stool to the left of the trash can.
The person approached the trash can.
The stool approached the trash can.
The person carried the stool.
The person carried the stool towards the trash can.
The stool approached the trash can to the left of the traffic cone.
The backpack to the left of the traffic cone approached the trash can.
The backpack to the right of the trash can approached the trash can.
The traffic cone approached the stool to the left of the trash can.
The trash can approached the chair.
The trash can to the left of the chair approached the chair.
The trash can approached the chair to the left of the backpack.
The person approached the chair.
The person picked up the trash can to the left of the stool.
The person approached the traffic cone.
The chair approached the traffic cone.
The person to the left of the backpack approached the traffic cone.
The person carried the chair towards the traffic cone.
The person put down the chair to the right of the backpack.
The person to the right of the traffic cone put down the chair.
The person to the right of the trash can put down the traffic cone.
The person to the left of the backpack put down the traffic cone.
The person put down the traffic cone slowly.
The person picked up the chair to the right of the backpack.
The person to the right of the trash can picked up the chair.
The stool approached the traffic cone to the right of the chair.
The stool approached the traffic cone to the left of the person.
The person picked up the traffic cone quickly.
The person picked up the traffic cone to the left of the stool.
The person to the left of the chair picked up the traffic cone.
The person picked up the backpack.
The person to the left of the chair picked up the backpack.
The person put down the backpack.
The person put down the backpack slowly.
The person to the right of the chair put down the backpack.
The person put down the backpack to the right of the trash can.
The traffic cone approached the stool.
The traffic cone to the left of the trash can approached the stool.
The traffic cone to the right of the stool approached the stool.
The backpack approached the trash can.
The backpack approached the trash can to the right of the stool.
The backpack to the right of the stool approached the trash can.
The person carried the chair.
The person to the left of the stool carried the chair.
The person carried the chair to the left of the traffic cone.
The person picked up the trash can.
The person picked up the trash can quickly.
The person picked up the trash can to the right of the stool.
The person picked up the traffic cone.
The person picked up the traffic cone slowly.
The person to the left of the stool picked up the traffic cone.
The person picked up the traffic cone to the right of the trash can.
The stool approached the traffic cone.
The stool to the left of the traffic cone approached the traffic cone.
The stool to the right of the chair approached the traffic cone.
The chair approached the trash can.
The chair to the left of the traffic cone approached the trash can.
The chair to the left of the trash can approached the trash can.
The person put down the stool.
The person to the left of the traffic cone put down the stool.
The traffic cone approached the chair to the left of the stool.
The traffic cone approached the chair.
The person carried the traffic cone towards the chair.
The person to the left of the stool carried the traffic cone.
The person carried the traffic cone away from the chair.
The person to the left of the stool put down the backpack.
The person put down the backpack to the right of the chair.
The person picked up the stool slowly.
The person to the right of the trash can put down the stool.
The traffic cone approached the trash can.
The traffic cone to the right of the stool approached the trash can.
The chair approached the stool to the left of the traffic cone.
The chair to the right of the stool approached the stool.
The person to the left of the stool put down the trash can.
The person put down the trash can to the left of the traffic cone.
The person approached the stool.
The backpack approached the stool.
The person carried the backpack towards the stool.
The backpack approached the chair.
The backpack to the right of the chair approached the chair.
The backpack to the right of the traffic cone approached the chair.
The person carried the stool away from the traffic cone.
The person to the left of the traffic cone picked up the backpack.
The stool approached the backpack.
The stool approached the backpack to the right of the trash can.
The stool to the left of the backpack approached the backpack.
The person to the left of the chair approached the stool.
The person carried the stool towards the chair.
The person to the left of the chair put down the stool.
The person put down the stool slowly.

Table 2: A selection of sentences (first part) drawn from the grammar in Table 11(a) that were used to annotate the clips for the second corpus.

The person to the right of the trash can approached the chair.
The person to the right of the trash can carried the chair.
The person to the right of the trash can put down the chair.
The person put down the chair slowly.
The person to the right of the trash can approached the stool.
The person picked up the stool to the right of the trash can.
The person put down the stool to the right of the trash can.
The person to the left of the stool approached the chair.
The person picked up the chair to the left of the stool.
The person carried the chair towards the stool.
The person to the left of the stool put down the chair.
The person to the right of the chair approached the trash can.
The person picked up the trash can to the right of the chair.
The person carried the trash can away from the chair.
The person put down the trash can to the right of the chair.
The person picked up the stool quickly.
The person put down the stool quickly.
The person approached the chair to the left of the stool.
The person put down the chair to the left of the stool.
The trash can approached the traffic cone.
The trash can to the right of the backpack approached the traffic cone.
The trash can approached the traffic cone to the right of the backpack.
The person to the right of the chair put down the trash can.
The person carried the chair towards the backpack.
The chair approached the backpack.
The chair approached the backpack to the left of the stool.
The person carried the trash can towards the traffic cone.
The person to the right of the backpack picked up the stool.
The person to the right of the backpack picked up the traffic cone.
The person to the left of the trash can picked up the traffic cone.
The trash can approached the stool.
The trash can to the left of the stool approached the stool.
The trash can to the right of the chair approached the stool.
The person picked up the trash can to the left of the chair.
The person carried the backpack away from the chair.
The person to the left of the traffic cone carried the backpack.
The person carried the stool away from the chair.
The person to the right of the chair picked up the backpack.
The person to the left of the trash can picked up the backpack.
The person picked up the backpack quickly.
The traffic cone approached the backpack.
The traffic cone to the left of the backpack approached the backpack.
The traffic cone approached the backpack to the left of the stool.
The person to the left of the traffic cone picked up the chair.
The person to the left of the trash can put down the traffic cone.
The person put down the traffic cone to the right of the stool.
The person carried the traffic cone towards the trash can.
The person carried the traffic cone away from the stool.
The stool approached the chair.
The stool approached the chair to the right of the traffic cone.
The stool to the right of the traffic cone approached the chair.
The person to the left of the traffic cone put down the backpack.
The person to the right of the trash can put down the backpack.
The chair to the left of the backpack approached the backpack.
The chair approached the backpack to the left of the trash can.

The person picked up the stool to the right of the traffic cone.
The person to the left of the stool picked up the trash can.
The person put down the stool to the left of the chair.
The person to the left of the trash can carried the stool.
The person put down the backpack quickly.
The person to the left of the backpack put down the chair.
The trash can to the left of the backpack approached the chair.
The person carried the trash can towards the chair.
The person picked up the chair slowly.
The person picked up the stool to the left of the chair.
The person picked up the backpack to the right of the trash can.
The person carried the trash can away from the backpack.

Table 3: A selection of sentences (second part) drawn from the grammar in Table 11(a). The sentences above the horizontal line were used to annotate the clips for the second corpus and those below it were used for test.

between nouns and object classes. Note, however, that since the training clips each contain more than one object, the task of learning noun meanings is still challenging. We filmed our own corpora as we are unaware of any existing corpora that exhibit the above properties.[2]

We annotated each of the 94 video clips in each corpus with human judgments. The first corpus was annotated against each of the 21 sentences from Table 1, indicating whether the given clip depicted the given sentence. Table 4 provides statistics on this annotation. The resulting set of $94 \times 21 = 1974$ judgments and the associated statistics were used to compare and contrast our machine-generated results against human judgments in the analyses in Sections 5.3 and 5.4. Each clip in the second corpus was used either for training or test, depending on the cross-validation fold, as described earlier. When it was included in the training set, it was paired with between 1 and 5 sentences selected from the 175

2. The video clips, sentential annotation described below, and all code needed to replicate the experiments in this section are available at `http://upplysingaoflun.ecn.purdue.edu/~qobi/cccp/grounding-language-in-video.html`.

	μ	σ
#Clips that depict a given sentence	12.33	6.48
#Sentences that describe a given clip	2.76	1.22

Table 4: Annotation statistics for the first corpus.

	μ	σ
#Clips that depict a given sentence	2.00	0.58
#Sentences that describe a given clip	0.37	0.61

Table 5: Annotation statistics for the second corpus.

training sentences from Tables 2 and 3 that were deemed to describe the associated training clip by a human judge. On average, each training clip was paired with 2.94 sentences. Collectively, the corpus contains 276 video-sentence pairs used for training. The three training folds contained 213, 208, and 204 video-sentence pairs respectively. When a given clip was included in the test set, it was paired with all 12 test sentences from Table 3. Thus the $94 - 29 = 65$ potential test clips in the second corpus were annotated against each of the 12 test sentences from Table 3, indicating whether the given clip depicted the given sentence. Table 5 provides statistics on this annotation. The resulting set of $65 \times 12 = 780$ judgments and the associated statistics were used to compare and contrast our machine-generated results against human judgments in the analyses in Section 5.5.

All of our experiments use an off-the-shelf object detector (Felzenszwalb et al., 2010a, 2010b) which outputs detections in the form of scored axis-aligned rectangles. In particular, we used the implementation described by Song, Zickler, Althoff, Girshick, Fritz, Geyer, Felzenszwalb, and Darrell (2012). Using off-the-shelf software, we trained six object detectors, one for each of the six object classes in our corpora: person, backpack, chair, traffic cone, trash can, and stool. To compensate for false negatives, as described in Section 2.1, we lowered the acceptance threshold on the models produced by automatic training. The per-part thresholds were uniformly reduced by 1.2, the model thresholds were uniformly reduced by 2.0, and non-maxima suppression was set to 0.6 for the first corpus and 0.55 for the second. We applied the person, backpack, chair, and trash can detectors uniformly to all frames of all video clips in the first corpus and all six detectors to all frames of all clips in the second corpus. For the first corpus, we selected the five highest-scoring detections produced by each object detector in each frame and pooled the results yielding twenty detections per frame. For the second corpus, we selected the two highest-scoring detections produced by each object detector in each frame and pooled the results yielding twelve detections per frame. While having a larger pool of detections per frame can better compensate for false negatives in object detection and potentially yield smoother tracks, it increases the size of the lattice and the concomitant running time but does not lead to appreciably better performance on our corpora.

5.2 The Manually-Constructed Lexicons

The experiments in Sections 5.3 and 5.4 use manually-constructed FSMs to represent word meanings when evaluating language inference and language generation. These hand-written representations of word meaning clearly encode pretheoretic human intuition and make such

intuition perspicuous. For these experiments, we formulate the word models for the lexical entries in Table 11(a) that appear in the sentences in Table 1. The experiments in Section 5.5 learn word models represented as HMMs. We evaluated these learned word models, in part, by comparison with manually-constructed HMMs. These manually-constructed HMMs will be discussed in Section 5.5.

We formulate the FSMs as regular expressions over predicates computed from detections. The particular set of regular expressions and associated predicates that are used in the experiments in Sections 5.3 and 5.4 are given in Table 6. The predicates are formulated around a number of primitive functions. The function *avgFlow*(b) computes a vector that represents the average optical flow inside the detection b. The function *model*(b) returns the object class of b. The function $x(b)$ returns the x-coordinate of the center of b. The function *hue*(b) returns the average hue of the pixels inside b. The function *angleSep* determines the angular distance between two angular arguments. The function *fwdProj*(b) displaces b by the average optical flow inside b. The function $\angle$ determines the angular component of a given vector. The function $\perp$ computes a normal unit vector for a given vector. The argument v to noJitter denotes a specified direction represented as a 2D unit vector in that direction. Predicates that take a single detection b as their sole argument can serve as 0/1 output models $h(k, b)$ ($-\infty/0$ in log space) for single-participant word models. Predicates that take a pair of detections b_1 and b_2 as their sole arguments can serve as 0/1 output models $h(k, b_1, b_2)$ ($-\infty/0$ in log space) for two-participant word models. Regular expressions are formulated around predicates as atoms. A given regular expression must be formed solely from output models of the same arity and denotes a word model with a 0/1 state-transition function ($-\infty/0$ in log space) where the output models are associated with the appropriate states.

5.3 Experiment 1: Language Inference

Tracking is traditionally performed using cues from motion, object detection, and/or manual initialization on an object of interest (Yilmaz, Javed, & Shah, 2006). However, in the case of a cluttered scene involving multiple events occurring simultaneously, there can be many moving objects, many instances of the same object class, and perhaps even multiple simultaneously occurring instances of the same event class. Here we illustrate how one can use a sentential description to guide the tracking of objects based on which ones participate in the target event.

The sentence tracker can focus its attention on just those objects that participate in an event specified by a sentential description. Such a description can differentiate between different simultaneous events taking place between many moving objects in the scene using descriptions constructed out of a variety of parts of speech. Using nouns to specify object class, one could differentiate between

> *The person picked up the backpack* and
> *The person picked up the chair.*

Using adjectives to specify object properties, one could differentiate between

> *The person picked up the red object* and
> *The person picked up the blue object.*

Constants

$\textsc{xBoundary} \triangleq 300\text{px}$ $\quad \textsc{nextTo} \triangleq 50\text{px}$ $\quad \Delta\textsc{static} \triangleq 6\text{px}$ $\quad \Delta\textsc{jump} \triangleq 30\text{px}$ $\quad \Delta\textsc{quick} \triangleq 80\text{px}$

$\Delta\textsc{slow} \triangleq 30\text{px}$ $\quad \Delta\textsc{closing} \triangleq 10\text{px}$ $\quad \Delta\textsc{angle} \triangleq 30°$ $\quad \Delta\textsc{hue} \triangleq 30°$

Simple Predicates

$$\textsc{noJitter}(b, v) \triangleq \|\mathit{avgFlow}(b) \cdot v\| \leq \Delta\textsc{jump} \qquad \textsc{alike}(b_1, b_2) \triangleq \mathit{model}(b_1) = \mathit{model}(b_2)$$
$$\textsc{close}(b_1, b_2) \triangleq |x(b_1) - x(b_2)| < \textsc{xBoundary} \qquad \textsc{far}(b_1, b_2) \triangleq |x(b_1) - x(b_2)| \geq \textsc{xBoundary}$$
$$\textsc{left}(b_1, b_2) \triangleq 0 < x(b_2) - x(b_1) \leq \textsc{nextTo} \qquad \textsc{right}(b_1, b_2) \triangleq 0 < x(b_1) - x(b_2) \leq \textsc{nextTo}$$
$$\textsc{hasColor}(b, \text{hue}) \triangleq \mathit{angleSep}(\mathit{hue}(b), \text{hue}) \leq \Delta\textsc{hue} \qquad \textsc{stationary}(b) \triangleq \|\mathit{avgFlow}(b)\| \leq \Delta\textsc{static}$$
$$\textsc{quick}(b) \triangleq \|\mathit{avgFlow}(b)\| \geq \Delta\textsc{quick} \qquad \textsc{slow}(b) \triangleq \|\mathit{avgFlow}(b)\| \leq \Delta\textsc{slow}$$
$$\textsc{person}(b) \triangleq \mathit{model}(b) = \textbf{person} \qquad \textsc{backpack}(b) \triangleq \mathit{model}(b) = \textbf{backpack}$$
$$\textsc{chair}(b) \triangleq \mathit{model}(b) = \textbf{chair} \qquad \textsc{trashcan}(b) \triangleq \mathit{model}(b) = \textbf{trashcan}$$
$$\textsc{blue}(b) \triangleq \textsc{hasColor}(b, 225°) \qquad \textsc{red}(b) \triangleq \textsc{hasColor}(b, 0°)$$

Complex Predicates

$$\textsc{stationaryClose}(b_1, b_2) \triangleq \textsc{stationary}(b_1) \wedge \textsc{stationary}(b_2) \wedge \neg\textsc{alike}(b_1, b_2) \wedge \textsc{close}(b_1, b_2)$$
$$\textsc{stationaryFar}(b_1, b_2) \triangleq \textsc{stationary}(b_1) \wedge \textsc{stationary}(b_2) \wedge \neg\textsc{alike}(b_1, b_2) \wedge \textsc{far}(b_1, b_2)$$
$$\textsc{closer}(b_1, b_2) \triangleq |x(b_1) - x(b_2)| > |x(\mathit{fwdProj}(b_1)) - x(b_2)| + \Delta\textsc{closing}$$
$$\textsc{farther}(b_1, b_2) \triangleq |x(b_1) - x(b_2)| < |x(\mathit{fwdProj}(b_1)) - x(b_2)| + \Delta\textsc{closing}$$
$$\textsc{moveCloser}(b_1, b_2) \triangleq \textsc{noJitter}(b_1, (0, 1)) \wedge \textsc{noJitter}(b_2, (0, 1)) \wedge \textsc{closer}(b_1, b_2)$$
$$\textsc{moveFarther}(b_1, b_2) \triangleq \textsc{noJitter}(b_1, (0, 1)) \wedge \textsc{noJitter}(b_2, (0, 1)) \wedge \textsc{farther}(b_1, b_2)$$
$$\textsc{inDirection}(b, v) \triangleq \textsc{noJitter}(b, \perp(v)) \wedge \neg\textsc{stationary}(b) \wedge \mathit{angleSep}(\angle\mathit{avgFlow}(b), \angle v) < \Delta\textsc{angle}$$
$$\textsc{approaching}(b_1, b_2) \triangleq \neg\textsc{alike}(b_1, b_2) \wedge \textsc{stationary}(b_2) \wedge \textsc{moveCloser}(b_1, b_2)$$
$$\textsc{departing}(b_1, b_2) \triangleq \neg\textsc{alike}(b_1, b_2) \wedge \textsc{stationary}(b_2) \wedge \textsc{moveFarther}(b_1, b_2)$$
$$\textsc{carry}(b_1, b_2, v) \triangleq \textsc{person}(b_1) \wedge \neg\textsc{alike}(b_1, b_2) \wedge \textsc{inDirection}(b_1, v) \wedge \textsc{inDirection}(b_2, v)$$
$$\textsc{carrying}(b_1, b_2) \triangleq \textsc{carry}(b_1, b_2, (0, 1)) \vee \textsc{carry}(b_1, b_2, (0, -1))$$
$$\textsc{pickingUp}(b_1, b_2) \triangleq \textsc{person}(b_1) \wedge \neg\textsc{alike}(b_1, b_2) \wedge \textsc{stationary}(b_1) \wedge \textsc{inDirection}(b_2, (0, 1))$$
$$\textsc{puttingDown}(b_1, b_2) \triangleq \textsc{person}(b_1) \wedge \neg\textsc{alike}(b_1, b_2) \wedge \textsc{stationary}(b_1) \wedge \textsc{inDirection}(b_2, (0, -1))$$

Regular Expressions

$$\lambda_{\mathit{person}} \triangleq \textsc{person}^+ \qquad \lambda_{\mathit{backpack}} \triangleq \textsc{backpack}^+ \qquad \lambda_{\mathit{chair}} \triangleq \textsc{chair}^+$$
$$\lambda_{\mathit{trash\ can}} \triangleq \textsc{trashcan}^+ \qquad \lambda_{\mathit{object}} \triangleq (\textsc{backpack} \mid \textsc{chair} \mid \textsc{trashcan})^+$$
$$\lambda_{\mathit{blue}} \triangleq \textsc{blue}^+ \qquad \lambda_{\mathit{red}} \triangleq \textsc{red}^+ \qquad \lambda_{\mathit{quickly}} \triangleq \textsc{true}^+\ \textsc{quick}^{[3,]}\ \textsc{true}^+$$
$$\lambda_{\mathit{to\ the\ left\ of}} \triangleq \textsc{left}^+ \qquad \lambda_{\mathit{to\ the\ right\ of}} \triangleq \textsc{right}^+ \qquad \lambda_{\mathit{slowly}} \triangleq \textsc{true}^+\ \textsc{slow}^{[3,]}\ \textsc{true}^+$$
$$\lambda_{\mathit{approached}} \triangleq \textsc{stationaryFar}^+\ \textsc{approaching}^{[3,]}\ \textsc{stationaryClose}^+$$
$$\lambda_{\mathit{carried}} \triangleq \textsc{stationaryClose}^+\ \textsc{carrying}^{[3,]}\ \textsc{stationaryClose}^+$$
$$\lambda_{\mathit{picked\ up}} \triangleq \textsc{stationaryClose}^+\ \textsc{pickingUp}^{[3,]}\ \textsc{stationaryClose}^+$$
$$\lambda_{\mathit{put\ down}} \triangleq \textsc{stationaryClose}^+\ \textsc{puttingDown}^{[3,]}\ \textsc{stationaryClose}^+$$
$$\lambda_{\mathit{towards}} \triangleq \textsc{stationaryFar}^+\ \textsc{approaching}^{[3,]}\ \textsc{stationaryClose}^+$$
$$\lambda_{\mathit{away\ from}} \triangleq \textsc{stationaryClose}^+\ \textsc{departing}^{[3,]}\ \textsc{stationaryFar}^+$$

Table 6: The FSMs representing the meanings of the lexical entries in Table 11(a) that appear in the sentences in Table 1 used for the experiments in Sections 5.3 and 5.4.

Using verbs to specify events, one could differentiate between

> *The person picked up the red object* and
> *The person put down the red object.*

Using adverbs to specify motion properties, one could differentiate between

> *The person quickly picked up the red object* and
> *The person slowly picked up the red object.*

Using prepositions to specify (changing) spatial relations between objects, one could differentiate between

> *The person to the right of the chair picked up an object* and
> *The person to the left of the chair picked up an object.*

Furthermore, such a sentential description can even differentiate which objects to track based on the role that they play in an event: agent, patient, source, goal, or referent. For example, the sentence *The person picked up the backpack to the left of the chair* differs from *The person picked up the chair to the left of the backpack* in that the roles of the backpack and the chair are exchanged. Although the same objects are involved in the described events, their roles in the events differ, and can be distinguished by the tracker. Figure 15 demonstrates this ability: different tracks are produced for the same video that depicts multiple simultaneous events when focused with different sentences. In this figure, as well as Figure 16 and Figures 21 and 22 in Appendix B, the boxes around the participants are color coded to indicate semantic role: agent in red, patient in blue, source in violet, goal in turquoise, and referent in green. This particularly illustrates that our system understands the image regions that correspond to the participants and the particular mapping of such to argument positions of predicates that denote the meanings of lexical items in the sentential description. This further illustrates deep semantic understanding.

Figure 15 evaluates this ability for each sentential position. Figure 21 in Appendix B evaluates this ability on all 9 minimal pairs, as indicated by the 'a' and 'b' variants of sentences 1–9 in Table 1, collectively applied to all 25 suitable video clips in the first corpus. We discard two clips from the original set of $9 \times 3 = 27$ video clips due to the fact that they involve an adjective (*grey*), corresponding to the chair, that cannot be reliably extracted from the video. For 18 out of the 25, both sentences in the minimal pair yielded track collections deemed to be correct depictions. We determine error from subjective human judgment of whether the track collection that our system produces matches the desired description. All of the errors encountered in this task fall into one of two categories. One category deals with the use of a color adjective along with the generic word *object* in the presence of some other entity in the video other than the intended object that incidentally has a similar color. The sole error in this category involves the tracker selecting detections on a person's red shirt instead of the red backpack, for one of three instances of minimal pair 2 in Table 1: *The red object approached the chair* and *The blue object approached the chair*. The correct result is obtained for the other instance of this minimal pair when associated with different video clips. The other category is largely due to the deficiencies of the detectors, particularly that for the *trash can*. In at least four instances, the paucity

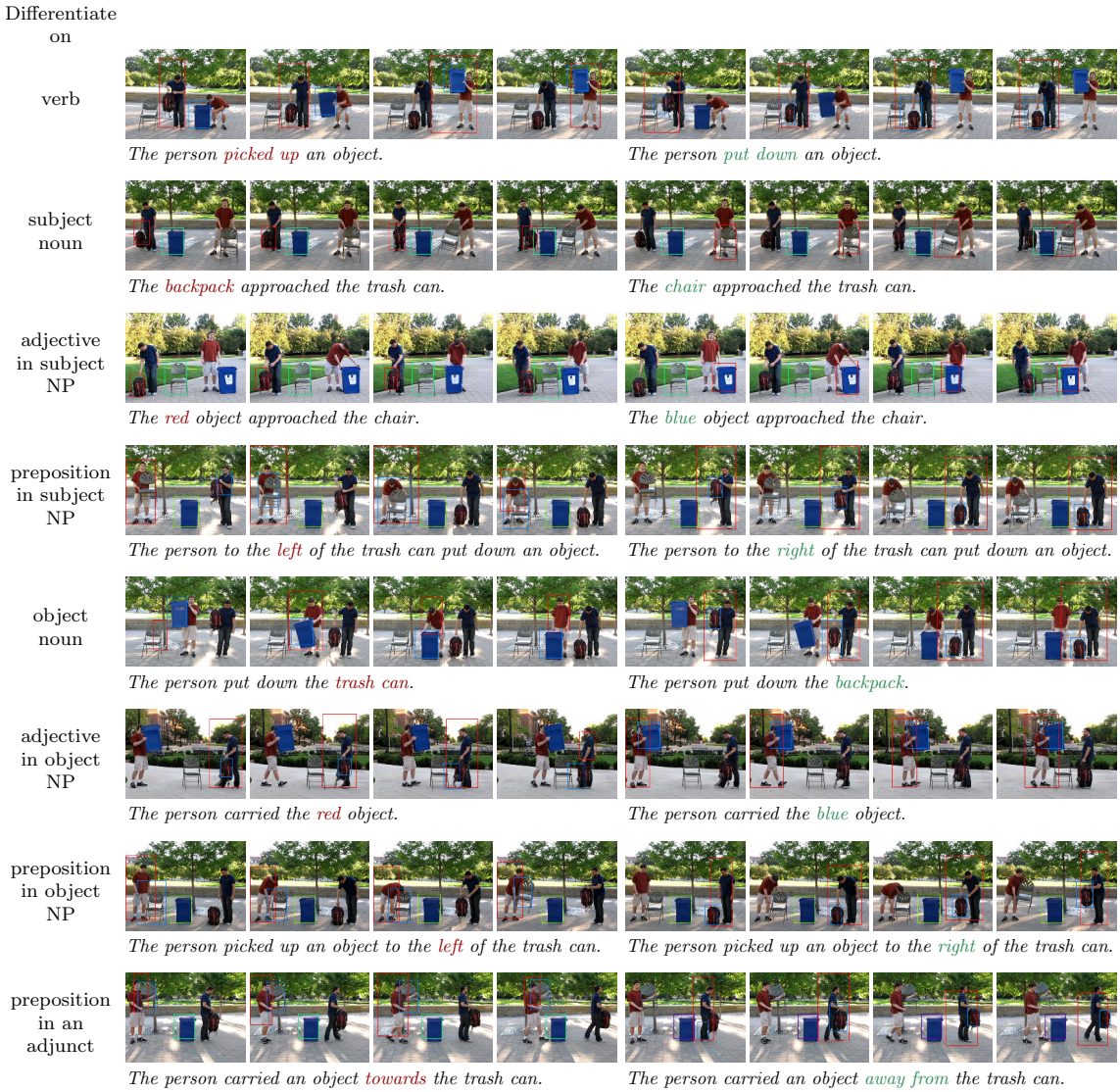


Figure 15: Language inference: two different track collections for the same video clip produced under guidance of two different sentences. Each clip is processed by a minimal pair, a sentence that varies in a single lexical item highlighted in red *vs.* green. The varying lexical item itself varies among all sentential positions across the eight examples. Results for all video clips processed by the minimal pairs in sentences 1–9 from Table 1 are included in Figure 21 in Appendix B. In this figure, as well as in Figures 16, 21, and 22, we indicate thematic role of the participants by the color of the bounding box: the red box denotes the agent, the blue box denotes the patient, the violet box denotes the source, the turquoise box denotes the goal, and the green box denotes the referent. These roles are determined automatically using the techniques in Appendix A.

Contraction Threshold	Accuracy
0.95	67.02%
0.90	71.27%
0.85	64.89%

Table 7: Accuracy as a function of contraction threshold.

of detections from the *trash can* detector results either in poor tracks or a complete failure to satisfy the FSMs corresponding to other word models. This is further exacerbated in the case of adverbs. Since adverbs modify verbs, and verbs vary in the manner of their execution, tight bounds on what would constitute *quickly* or *slowly* are difficult to obtain. Any bounds we are able to impose are sufficiently noisy that sometimes the distinction between an action happening *quickly* or *slowly* is lost. Two such errors occur here, namely on two instances of minimal pair 8 in Table 1: *The person picked up an object quickly* and *The person picked up an object slowly*. The correct result is obtained for the remaining instance of this minimal pair when associated with a different video clip.

5.4 Experiment 2: Language Generation

We can use the ability of the sentence tracker to score a video-sentence pair to generate a sentence that describes a given video clip by searching for the highest-scoring sentence for that clip. However, this has a problem. Recall that f, g, h, and a are all values in log space that range in $(-\infty, 0]$ where increasing value denotes higher score, *i.e.,* better fit to the model. Since the sentence-tracker scoring function (Equation 10) sums these, scores decrease with longer word strings and greater numbers of participants that result from longer word strings. So we don't actually search for the highest-scoring sentence, which would bias the process towards short sentences. Instead we seek complex sentences that describe the clip as they are more informative.

Nominally, this search process would be intractable since the space of possible sentences can be huge and even infinite. However, we can use beam search to get an approximate answer. This is possible because the sentence tracker can score any word sequence, not just complete sentences, as long one can construct a linking function θ. We can select the top-scoring single-word sequences and then repeatedly extend the top-scoring W-word sequences, by one word, to select the top-scoring $W+1$-word sequences, subject to the constraint that a linking function θ exists for these $W+1$ words and these $W+1$-word sequences can be extended to grammatical sentences by insertion of additional words. We terminate the search process when the *contraction threshold*, the ratio between the score of a sequence and the score of the sequence expanding from it, drops below a specified value and the sequence being expanded is a complete sentence. This contraction threshold controls complexity of the generated sentence.

When restricted to FSMs, h and a will be $0/1$ which become $-\infty/0$ in log space. Thus increase in the number of words can only decrease a score to $-\infty$, meaning that a sequence of words no-longer describes a video clip. Since we seek sentences that do, we terminate the above beam-search process before the score goes to $-\infty$. In this case, there is no approximation: a beam search maintaining all W-word sequences with finite score yields the highest-scoring sentence before the contraction threshold is met.

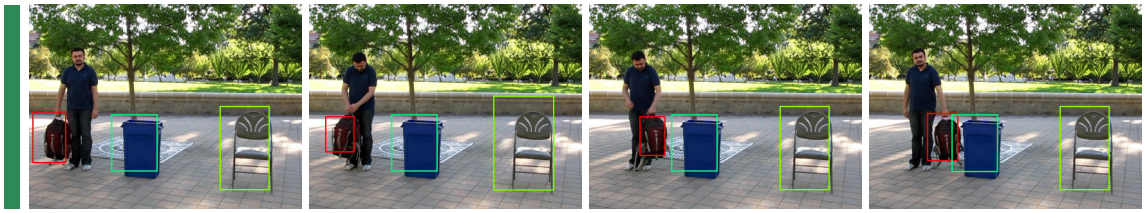
To evaluate this approach, we searched the space of sentences generated by the grammar in Table 11(a) to find the top-scoring sentence for each of the 94 video clips in the first corpus. Note that the grammar generates an infinite number of sentences due to recursion in NP. Even restricting the grammar to eliminate NP recursion yields a space of 147,123,874,800 sentences. Despite not restricting the grammar in this fashion, we are able to effectively find good descriptions of the video clips.

We evaluated the accuracy of the sentence tracker in generating descriptions for all 94 video clips in the first corpus for multiple contraction thresholds. Accuracy was computed as the percentage of the 94 clips for which the sentence tracker produced descriptions that were deemed to describe the video by human judges. The resulting accuracy for different contraction thresholds is shown in Table 7. Figure 16 shows the highest-scoring sentence generated by this approach for several clips in the first corpus for the contraction threshold 0.90. Figure 22 in Appendix B shows the highest-scoring sentence generated by this approach on each of the 94 clips in the first corpus. To illustrate the effect of the contraction threshold, we show below, the generated sentence for the corresponding contraction thresholds for the first video clip in Figure 16.

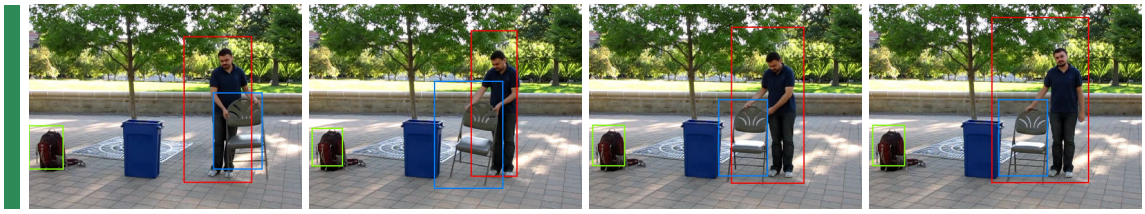
0.95 *The backpack approached the trash can.*
0.90 *The backpack to the left of the chair approached the trash can.*
0.85 *The backpack to the left of the chair approached the trash can.*

An important distinction between this approach and the state of the art for generating sentential video description is the generativity of the labeling domain. In existing work (Kulkarni et al., 2011; Gupta, Verma, & Jawahar, 2012), the process of labeling events in video involves searching for phrases or sentences that best match the video using a trained set of classifiers. This process usually involves extracting correspondences between labels and video features in a training corpus. The training corpus labels each video with a word or phrase and the sentence-generation process on an unseen video labels that video either with an existing label from the training corpus or a simple concatenation of such labels. In contrast, our approach can label an unseen video with any grammatical utterance admitted by the grammar and lexicon, from a potentially unbounded set, even ones that have never appeared, in whole or in part, in the training set.

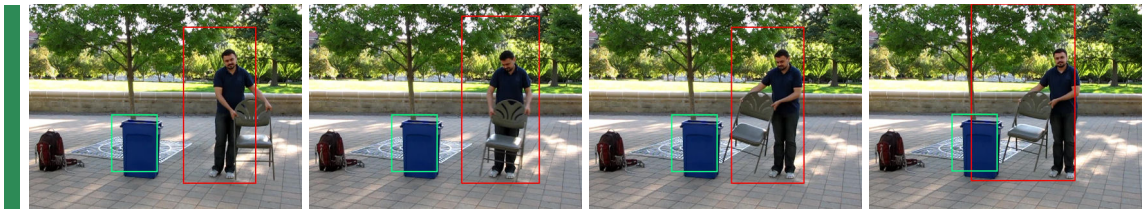
The sentence-tracker framework can also generate sentential video description from a fixed set of sentential labels, simply by scoring each potential label against an unseen video clip and selecting the top-scoring label. We evaluate this ability by labeling each of the 94 video clips in the first corpus from the fixed label set of 21 sentences shown in Table 1 and comparing such with human judgments. We performed three analyses. First, we measured the percentage of clips that depict their top-scoring sentence as determined by human judges. This was determined to be 94.68%. Chance performance is 13.12%, since on average, 2.76 sentences are deemed to describe a given clip, as shown in Table 4. Second, if we relax our selection criterion slightly, to consider the percentage of clips described by at least one of the top-three sentences, we obtain 100% accuracy. Chance performance is $1 - (1 - 0.1312)^3 \approx 34.42\%$. Finally, we can threshold the video-sentence score, yielding a binary machine judgment as to whether a given sentence describes a given clip, or alternatively whether a given clip depicts a given sentence. We can then ask how well such machine judgments match human judgment over all $94 \times 21 = 1974$ video-sentence pairs in the first corpus.



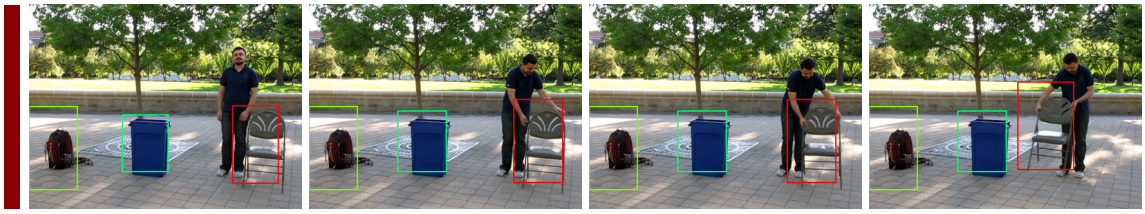
The backpack to the left of the chair approached the trash can.



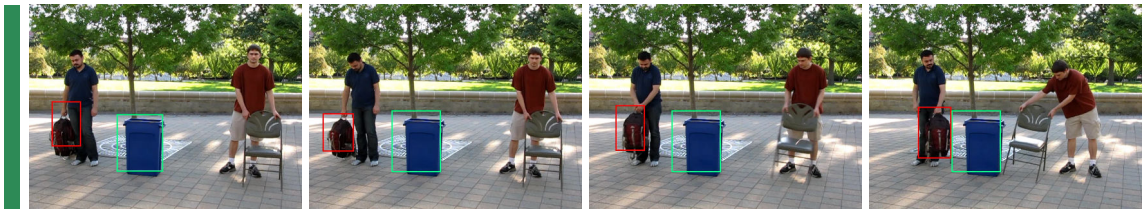
The person to the right of the backpack carried the chair.



The person to the right of the trash can approached the trash can.



The chair to the right of the person approached the trash can.



The backpack to the left of the trash can approached the trash can.

Figure 16: Sentential descriptions generated for several video clips in the first corpus subject to the contraction threshold 0.90. The highest-scoring sentence for each clip is generated, among all sentences that are generated by the grammar in Table 11(a), by means of a beam search. The sentences deemed by human judges to describe the associated clips are indicated in green, while ones that do not are indicated in red. Sentential descriptions generated for each of the 94 video clips in the first corpus are shown in Figure 22 in Appendix B.

Searching for the threshold that maximizes this accuracy yields an accuracy of 86.88%. Chance performance is 13.12%, since 259 out of the 1974 human judgments are positive. Thus the sentence tracker performs significantly above chance on all three analyses.

5.5 Experiment 3: Language Acquisition

The sentence tracker, when wrapped in EM, can learn a lexicon that maps words to their meanings from a training set of video clips paired with sentences. A crucial distinction between this approach and the prior state of the art in learning object and event recognizers from video is that, in this approach, the training videos are paired with entire sentences, not individual class labels. These sentential labels are generative; the set of possible labels is infinite as they are generated by a context-free grammar that contains recursion. Thus the vast majority of the potential labels never appear in the training set. Yet our method can learn to describe previously unseen videos with previously unseen sentential labels that are composed of words that likely do not occur in a single training sample but instead require composing words that are each learned by exposure to distinct training samples.

To evaluate the use of the sentence tracker to perform language acquisition, we employ the second corpus described in Section 5.1, in particular Tables 2 and 3, together with the grammar and lexicon from Table 11. This language fragment contains 17 lexical entries over 6 parts of speech (1 determiner, 6 nouns, 2 spatial-relation prepositions, 4 verbs, 2 adverbs, and 2 motion prepositions). We model and learn the meanings of all the content words in this lexicon. Table 8 specifies the arity I, the number K of states, the feature-vector length N, the number Z of bins fore each feature, and the feature computation Φ for the word models of each part of speech c. While we specify a different subset of features for each part of speech, we presume that, in principle, with enough training data, we could include all features in all parts of speech and automatically learn which ones are noninformative and lead to uniform distributions.

We compute continuous features, such as velocity, distance, size ratio, and x-position from the detections and quantize the features into bins as follows:

velocity To reduce noise, we compute the velocity of a participant by averaging the optical flow in the detection. The velocity magnitude is quantized into 5 levels. For expository clarity, we refer to these levels mnemonically as absolutely stationary, mostly stationary, moving slowly, moving quickly, and moving very quickly. The velocity orientation is quantized into 4 directions: leftward, upward, rightward, and downward.

distance We compute the Euclidean distance between the detection centers of two participants, which is quantized into 3 levels: near, moderate distance, and far.

size ratio We compute the ratio of the detection area of the first participant to the detection area of the second participant, quantized into 2 levels: larger than and smaller than.

x-position We compute the difference between the x-coordinates of the participants, quantized into 2 levels: to the left of and to the right of.

The binning process was determined by a preprocessing step that clustered a subset of the training data. In addition to the above continuous features that need quantization, we also incorporate the index of the detector that produced the detection as a discrete feature.

c	I	K	N	Z	Φ
N	1	1	1	6	detector index
V	2	3	6	5	velocity magnitude for the first argument
				4	velocity orientation for the first argument
				5	velocity magnitude for the second argument
				4	velocity orientation for the second argument
				3	distance between the first and second arguments
				2	size of the first argument / size of the second argument
P	2	1	1	2	difference between the x-positions of the first and second arguments
Adv	1	3	1	5	velocity magnitude
P_M	2	3	2	5	velocity magnitude for first argument
				3	distance between the first and second arguments

Table 8: Characteristics of the HMMs used to model word meanings for various parts of speech c. I denotes arity, K denotes the number of states, N denotes the number of features in the output model, Z denotes number of bins for a particular feature, and Φ denotes the feature computation.

The detector index is mainly used for identifying a detection when learning nouns. The particular features computed for each part of speech are given in Table 8.

Note that while we use English phrases, like to the left of, to refer to particular bins of particular features, and we have object detectors which we train on samples of a particular object class such as backpack, such phrases are only mnemonic of the clustering and object-detector training process. We do not have a fixed correspondence between the lexical entries and any particular feature value. Moreover, that correspondence need not be one-to-one: a given lexical entry may correspond to a (time variant) constellation of feature values and any given feature value may participate in the meaning of multiple lexical entries.

We performed three-fold cross validation using the partitioning described in Section 5.1. It is important to stress that for each fold, the test set was disjoint from the training set, *both in video clips and in sentential labels*. This crucially allowed us to evaluate the *generative* nature of the sentential labels: the ability to learn to generate previously unseen labels for previously unseen video.

For each fold, we trained a lexicon on the training set for that fold using the procedure from Section 4. We then evaluated the trained lexicon on the test set for that fold by performing three distinct analyses:

1. comparing F1 score on the test set with a variety of baselines
2. comparing an ROC curve on the test set with a variety of baselines
3. inspection of the learned models and comparison with hand-constructed models

The first two analyses require scoring unseen video-sentence pairs. These could be scored with Equation 10. However, this score depend on the sentence length W, the length T of the video clip, the number L of participants, and the collective numbers of states K and feature-vector lengths N for the word models for words in that sentence. One can remove

Fold	Baselines			Our method
	Chance	Blind	Hand	
1	0.06	0.10	0.73	0.56
2	0.07	0.12	0.65	0.50
3	0.04	0.08	0.50	0.31
average	**0.06**	**0.10**	**0.62**	**0.46**

Table 9: A comparison of the F1 scores on the test sets between our method and a variety of baselines.

the dependence on the number L of participants by using $\mathbf{L}(\mathbf{B}; \mathbf{s}, \Lambda)$ as the score. However, this does not remove dependence on the other factors.

To render the scores comparable across such variation, we apply a sentence-length prior $\pi(\mathbf{s})$ to the average per-frame score computed from the whole-video score $\mathbf{L}(\mathbf{B}; \mathbf{s}, \Lambda)$:

$$[\mathbf{L}(\mathbf{B}; \mathbf{s}, \Lambda)]^{\frac{1}{T}} \pi(\mathbf{s})$$

where

$$\pi(\mathbf{s}) = \exp \sum_{w=1}^{W} \left[\omega(K_{s_w}) + \sum_{n=1}^{N_{s_w}} \omega(Z_{s_w}^n) \right]$$
$$\omega(Z) = -\sum_{z=1}^{Z} \frac{1}{Z} \log \frac{1}{Z} = \log Z$$

In the above, $\omega(Z)$ is the entropy of a uniform distribution over Z bins. This prior prefers longer sentences which are more descriptive of the video.

The resulting scores are thresholded to decide hits, which together with the manual annotation, can generate True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) counts. To conduct our first analysis, for each fold, we selected the threshold that led to the maximal F1 score on the training set, and used this threshold to compute the F1 score on the test set. Table 9 reports the per-fold F1 scores along with the average across folds.

For comparison, we also report F1 scores for three baselines: **Chance**, **Blind**, and **Hand**. The **Chance** baseline randomly classifies a video-sentence pair as a hit with probability 0.5. The **Blind** baseline determines hits by potentially looking at the sentence but never looking at the video. This strategy will make the same decision on video-sentence pairs if these pairs contain the same sentence. We can find an upper bound of the F1 score that a blind method could have on each of our test sets by solving a 0/1 fractional-programming problem as follows. An optimal blind baseline will try to find a decision d_m for each of the M test sentences $\mathbf{s}_m$ that maximizes the F1 score. Suppose, comparison with ground-truth yields FP_m false positives and TP_m true positives on the test set when

$d_m = 1$. Also suppose that setting $d_m = 0$ yields FN_m false negatives. The F1 score is then:

$$\frac{1}{\underbrace{1 + \frac{\sum_{m=1}^{M} d_m \mathrm{FP}_m + (1 - d_m)\mathrm{FN}_m}{\sum_{m=1}^{M} 2 d_m \mathrm{TP}_m}}_{\Delta}}$$

Thus to maximize F1 we seek to minimize the term Δ. This is an instance of 0/1 fractional-programming problem which can be solved by binary search or Dinkelbach's (1967) algorithm. This yields the best possible F1 score that any blind algorithm can produce. The **Hand** baseline determines hits with the hand-crafted HMMs described below. These were carefully designed to yield what we believe is near-optimal performance. As can be seen from Table 9, our trained word models perform substantially better than the **Chance** and **Blind** baselines and approach the performance of the **Hand** baseline. Because the corpus was counterbalanced, the **Chance** and **Blind** baselines exhibit similar poor performance.

To conduct our second analysis, we varied the threshold used to decide hits to produce ROC curves. Figure 17 shows curves for each of the folds along with an average across folds, comparing our trained word models against the various baselines. Again, our trained word models significantly outperform the baselines and essentially match the performance of the hand-crafted word models.

Good F1 scores and ROC curves are necessary but not sufficient to demonstrate successful learning. It is possible that the trained word models reflect artifactual properties of the corpus and don't encode the natural pretheoretic intended meaning. For example, if the dataset has spurious unintended correlations, such as whenever *approach* happens, the agent is always larger than the goal, the learned word model may reflect that correlation and this correlation may be the primary factor leading to good performance on the test set. If such an artifactual correlation is overly strong, it could even overpower the correlations between the relevant features and allow learning meanings that do not rely on those features and which would fail to generalize to corpora that did not exhibit the same artifactual properties.

To evaluate whether this occurs in our experiments, we conducted a third analysis that compared our trained word models (for fold 2) with the hand-crafted ones illustrated in Figures 23 through 30 in Appendix B. For qualitative comparison, we render the hand-crafted and trained word models side by side for each lexical entry, graphically illustrating the output distributions and textually illustrating the initial-state and state-transition-function distributions. Qualitative inspection indicates that the corresponding word models are indeed quite similar except for noise in the learned word models. The crucial qualitative observation is that to a large extent the initial-state and state-transition-function distributions place the bulk of the probability mass in the same state and the relevant output distributions exhibit peaks at the same bins. For example, for the word *person*, the two word models have a peak for the first bin which denotes the object-detector class person. Similarly, both word models for the verb *approached* describe the qualitative motion profile. Both depict an initial state in which:

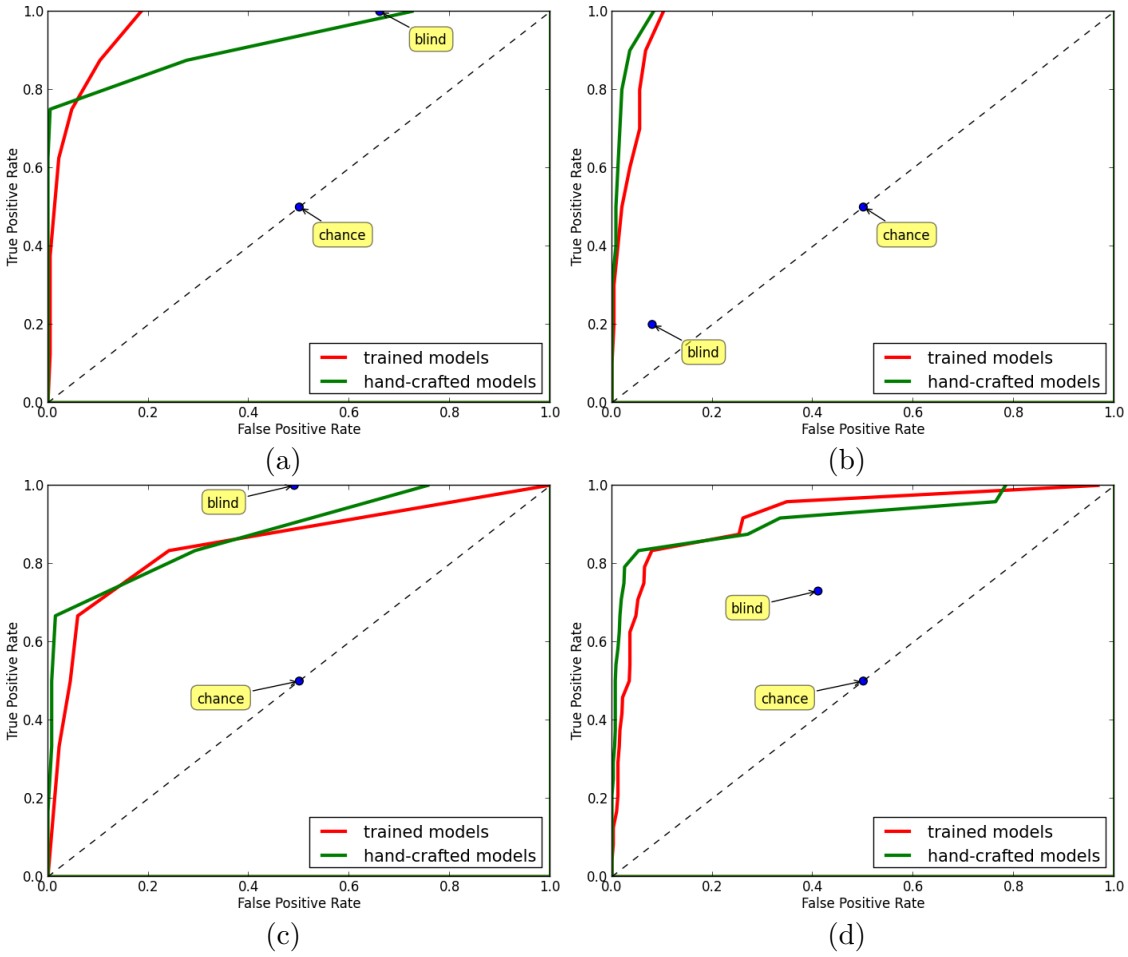


Figure 17: ROC curves comparing the performance of the trained models against the various baselines for the three folds (a-c) and averaged across fold (d).

1. the agent and the goal are both stationary and
2. the agent is far from the goal

followed by an intermediate state in which:

1. the agent is moving horizontally,
2. the goal is stationary, and
3. the distance between the participants is decreasing

followed by a final state in which:

1. the agent and the goal are both stationary and
2. the agent is close to the goal.

There are two primary qualitative differences between the learned and hand-crafted distributions. The first is noise. The second is that the hand-crafted distributions for irrelevant features are intentionally uniform while the learned distributions for these features sometimes encode the artifactual properties of the corpus to a small extent. For example, the second state of the trained word model for *picked up* indicates that the first argument is

	trained word models				random word models			
	1	2	3	**average**	1	2	3	**average**
person	0.00	0.00	0.00	**0.00**	1.11	1.09	3.45	**1.88**
backpack	0.00	0.00	0.00	**0.00**	5.43	1.72	1.14	**2.76**
chair	0.00	0.00	0.00	**0.00**	4.17	1.44	1.64	**2.42**
traffic cone	0.00	0.00	0.00	**0.00**	2.09	1.47	1.78	**1.78**
trash can	0.00	0.00	0.00	**0.00**	0.82	1.35	1.09	**1.09**
stool	0.00	0.00	0.00	**0.00**	1.12	1.33	4.10	**2.18**
to the left of	0.00	0.00	0.00	**0.00**	1.19	0.26	0.59	**0.68**
to the right of	0.00	0.00	0.00	**0.00**	0.50	0.09	0.53	**0.37**
approached	12.63	15.43	12.44	**13.50**	11.32	18.92	18.10	**16.11**
carried	15.89	10.60	11.74	**12.74**	14.42	11.97	15.10	**13.83**
picked up	9.40	9.44	10.97	**9.94**	12.86	8.49	14.44	**11.93**
put down	8.73	13.09	10.05	**10.62**	16.59	11.87	14.02	**14.16**
towards	1.71	4.69	3.14	**3.18**	3.97	3.88	4.65	**4.17**
away from	3.21	6.72	2.86	**4.27**	10.91	5.32	9.81	**8.68**

Table 10: An upper bound on the KL-divergence between the hand-crafted and trained word models for each fold and averaged across folds. (left) KL-divergence between trained word models and hand-crafted word models. (right) KL-divergence between random word models and hand-crafted word models.

moving upward while the hand-crafted word model contains a uniform distribution for the velocity orientation of the first argument. Similarly, the second and third states for the trained word model for *carried* appear, at first glance, to be quite different from the hand-written one. However, closer inspection reveals that they encode similar information. The second state in the hand-written word model actually corresponds to the last two states in the trained word model, which collectively encode a mixture distribution. The mixture distribution encodes that fact that *carried* is bidirectional and can involve leftward or rightward motion. The hand-written word model encodes this with a single state and a bimodal output distribution while the trained word model encodes this with two states each with unimodal output distributions. The lack of an additional state forces the trained word model to merge the output distributions for the velocity features from the last state in the hand-crafted word model into the two states that code the mixture distribution. We expect such differences to be eliminated with a larger training set or more accurate feature extraction.

We augmented this qualitative analysis of the similarity between the hand-crafted and trained word models with a quantitative analysis. We computed the KL-divergence between the output distributions of corresponding word models. This is not the true KL-divergence between two word models, as it ignores the initial-state distributions and state-transition functions, but provides a loose lower bound on the actual KL-divergence. Table 10 reports these for each word in our lexicon. Across the board, the trained word models are much closer to the hand-trained ones than the random word models.

6. Related Work

The language-inference task discussed in Section 5.3 requires a mechanism to focus attention on a particular activity in a video that depicts multiple simultaneous activities. Obtaining such a capability by extension of other state-of-the-art methods that can identify activity in video is not trivial. A large portion of such work, such as recently done by Kuehne et al. (2011) and Sadanand and Corso (2012), identify either a single activity in a given video or a rank ordering of possible activities. If such videos depicted multiple simultaneous identical activities, then these methods would identify only a single instance of such activity. This is partly due to the fact that matching features, say from STIP (Laptev, 2005), only provides a score, but no means of localization. Our method, on the other hand, can do so. If there exist two instances of an activity, say *pick up*, occurring simultaneously, we can specify which one to focus attention on by means of other elements in the video, such as characteristics of the participants (adjectives), manner of the action (adverbs), or relations between the participants and other *unrelated* objects in the scene (prepositions). As discussed previously in Section 5.4, much of the prior work on generating sentences to describe images (Jie, Caputo, & Ferrari, 2009; Farhadi, Hejrati, Sadeghi, Young, Rashtchian, Hockenmaier, & Forsyth, 2010; Kulkarni et al., 2011; Li & Ma, 2011; Yang, Teo, Daumé III, & Aloimonos, 2011; Gupta et al., 2012; Mitchell, Dodge, Goyal, Yamaguchi, Stratos, Han, Mensch, Berg, Berg, & III, 2012) and video (Kojima, Tamura, & Fukunaga, 2002; Fernández Tena, Baiget, Roca, & Gonzàlez, 2007; Barbu et al., 2012a; Hanckmann et al., 2012; Khan & Gotoh, 2012; Krishnamoorthy et al., 2013; Wang, Guan, Qiu, Zhuo, & Feng, 2013) uses special-purpose natural-language-generation methods. Our method, in contrast, systematically searches for the highest-scoring sentence generated by a grammar using the same video-sentence scoring function as used for language inference and language acquisition. The generativity of our labeling domain allows us to label an unseen video with any sentence, from a potentially unbounded set, including those that have never appeared, in whole or in part, in any form of training.

There has been active research on grounded language learning in the computational linguistics community. Some of this research employs approaches that directly map words to perceptual features extracted from the external world. Roy (2002) paired training sentences with vectors of real-valued features extracted from synthesized images which depict 2D blocks-world scenes, to learn a specific set of features for adjectives, nouns, and adjuncts. Roy and Pentland (2002) presented a computational model which acquires word meanings directly from multimodal sensory input. Yu and Ballard (2004) paired training images containing multiple objects with spoken name candidates for the objects to find the correspondence between lexical items and visual features. Marocco, Cangelosi, Fischer, and Belpaeme (2010) grounded the meanings of action words in the link between a robot's action effects and the behavior observed on the manipulated objects before and after the action. Because these approaches directly learn word meanings from associated features, they can only robustly understand a limited set of sentential fragments and lack the capability to deal with complex syntactic structures, since the resulting word meanings are neither generative nor compositional.

Other work within the computational linguistics community has focused on learning symbolic representations of word meanings from corpora of sentences paired with sym-

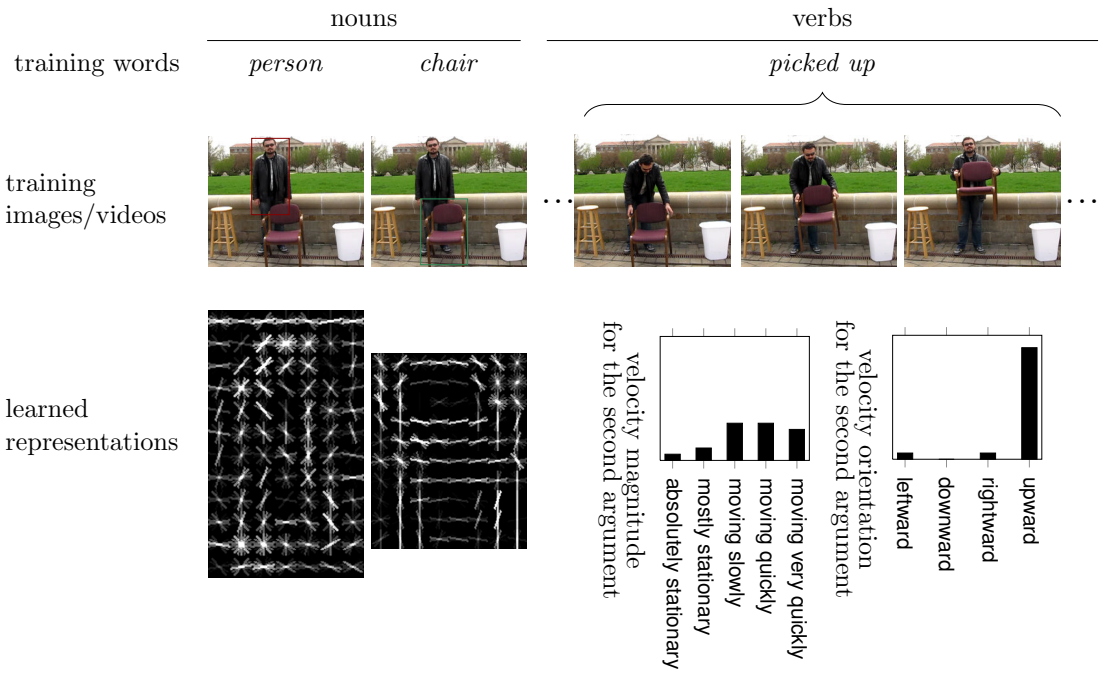
bolic representations of sentential meaning, as illustrated in Figure 18(a). Thompson and Mooney (2003) described a system called WOLFIE that acquires a semantic lexicon of phrase-meaning pairs from a corpus of sentences paired with semantic representations. Zettlemoyer and Collins (2005) presented a method for learning sentence meanings in the form of lambda-calculus encodings. Dominey and Boucher (2005) paired narrated sentences with symbolic representations of their meanings, automatically extracted from video, to learn object names, spatial-relation terms, and event names as mappings from the grammatical structure of sentential fragments to the semantic structure of the associated meaning representation. Piantadosi, Goodman, Ellis, and Tenenbaum (2008) employed an unsupervised, cross-situational Bayesian learning model for the acquisition of compositional semantics, to solve the problem of referential uncertainty. Chen and Mooney (2008) and Kim and Mooney (2010) learned the language of sportscasting by determining the alignment between game commentaries and the meaning representations output by a rule-based simulation of the game. This was later reduced to the task of learning a Probabilistic Context-Free Grammar (PCFG) by Börschinger, Jones, and Johnson (2011). Their subsequent work (Chen & Mooney, 2011; Kim & Mooney, 2012, 2013) proposed techniques for learning to follow navigation instructions from observation given weak, ambiguous supervision. Kwiatkowski, Zettlemoyer, Goldwater, and Steedman (2010) and Kwiatkowski et al. (2012) presented an approach that learns Montague-grammar representations of word meanings together with a combinatory categorial grammar (CCG) from child-directed sentences paired with first-order formulas that represent their meaning. Although these methods succeed in learning word meanings from sentential descriptions, they do so only for symbolic representations that might be extracted from simple or synthesized visual input; they fail to bridge the gap between language and computer vision, *i.e.,* they do not extract meaning representations from complex visual scenes.

More recent work in the computational linguistics and robotics communities has attempted to learn grounded word meanings from richer perceptual input paired with multi-word phrases. Krishnamurthy and Kollar (2013) introduced the Logical Semantics with Perception (LSP) framework for grounded language acquisition that learns to map natural language statements to their referents in a physical environment. However, they did this only for nouns and spatial-relation prepositions on a small set of static images. Tellex, Thaker, Joseph, and Roy (2013) learned the mapping between specific phrases and aspects of the external world for a robotic system, but they assumed an ideal scene: perfect object classification, a 3D coordinate system, and unambiguous demonstration of the robot correctly executing the action in the environment.

There has also been research on training object and event models from large corpora of complex images and video in the computer vision community (Feng, Manmatha, & Lavrenko, 2004; Yao, Yang, Lin, Lee, & Zhu, 2010; Kulkarni et al., 2011; Ordonez, Kulkarni, & Berg, 2011; Kuznetsova, Ordonez, Berg, Berg, & Choi, 2012; Sadanand & Corso, 2012; Chen & Grauman, 2013; Everts, van Gemert, & Gevers, 2013; Song, Morency, & Davis, 2013; Tian, Sukthankar, & Shah, 2013a), as illustrated in Figure 18(b). These can be viewed as learning meanings for nouns and verbs. However, most such work requires training data that labels individual concepts with individual words (*i.e.,* objects delineated via bounding boxes in images as nouns and events that occur in short video clips as verbs). In other words, they have to specify the correspondence between the concepts in the data

training sentence	*The person picked up the chair.*	
training meaning	CAUSE(PERSON, GO(CHAIR, UP))	
learned representations	*person*	PERSON
	picked up	CAUSE(x,GO(y,UP))
	chair	CHAIR

(a)



(b)

Figure 18: An illustration of the dominant paradigms in prior work. (a) Most work in the computational linguistics community learns symbolic representations of word meanings from sentences paired with symbolic representations of sentential meanings. (b) Most work in the computer vision community learns each word independently, from training data that annotates which image or video portion corresponds to an object or event label, with distinct representations for each part of speech.

and the words to be trained. There is no attempt to model phrasal or sentential meaning, let alone acquire the object or event models from training data labeled with phrasal or sentential annotation. As a result, the learned word meanings are neither generative nor compositional. Descriptions of new images and video are produced by mosaicing together previously learned sentence fragments. Moreover, unlike the methods presented here, these approaches use distinct representations for different parts of speech; *i.e.,* object and event recognizers use different representations.

Our method differs from prior work in three ways. First, our input consists of realistic video filmed in an outdoor environment. Second, we learn the entire lexicon, including nouns, verbs, adverbs, and prepositions, simultaneously from video described with *whole*

sentences. Third, we adopt a uniform representation for the meanings of words in all parts of speech, namely hidden Markov models (HMMs) whose states and distributions allow multiple possible interpretations of a word or a sentence in an ambiguous perceptual context.

The work presented here is most similar to three very recent papers (Das, Xu, Doell, & Corso, 2013; Rohrbach, Qin, Titov, Thater, Pinkal, & Schiele, 2013; Guadarrama, Krishnamoorthy, Malkarnenkar, Venugopalan, Mooney, Darrell, & Saenko, 2013) which generate text descriptions of video. On the surface, these papers appear to describe approaches that handle unrestricted text and video. However, deeper analysis reveals that this is not the case. Indeed, such analysis demonstrates that the space of text supported by these systems is far more restrictive than what we present here. We discuss this prior work in depth below along with such analysis.

Das et al. (2013) generate text descriptions of cooking videos garnered from YouTube. They do so by using shallow vision features on an unseen video to index into a training corpus of videos paired with text annotations to find similar videos and stitching together fragments of the text associated with the indexed videos to obtain a new text annotation of the unseen video.

1. It does not have a model of word or sentence meanings. It doesn't know what the words or the sentences in the annotations refer to in the video. One can't point to any component in the system and say this is its definition for this particular word. This is precisely what we do in Table 6 and Figures 5, 6, and 23–30 (in Appendix B). Moreover, one can't analyze what portions of the meanings are correct and what are wrong, as we do on page 51. When the system generates an incorrect annotation, there is nothing much one can say about it other than it did so.
2. Because of (1) it can't do what we call inference. It can't process a video with simultaneous actions taking place with different subsets of actors and objects in the video with two different sentences and highlight the different sets of participants for the different sentences. This is precisely what we demonstrate in Figure 15 and Figure 21 in Appendix B. This demonstrates deep understanding. The fact that we do so for minimal pairs, pairs of sentences that differ in a single word, and vary that word over all lexical entries and all sentential positions demonstrates that our semantic model reflects deep understanding of every word.
3. The work of Das et al. (2013) lacks such. This causes their method to generate a huge number of erroneous descriptions. Das et al. (2013, Figure 6) show five sample sentences generated for each of five sample videos. For all sample videos, between three and four of the generated sentences are false of the video. Most have completely incorrect objects and actions. These are the examples picked to showcase their system. Presumably, it performs worse on other examples. In contrast, we conduct and present results of a thorough evaluation: Figure 21 in Appendix B presents results on all examples, without exception.
4. Most of the nouns and adjectives generated in sentences in the work of Das et al. (2013, Figure 6) describe objects that are far beyond the ability of state-of-the-art object-detection systems to detect (e.g., *knob*, *pliers pieces of metal*, *glass bowl*, *porcelain bowl*, *sponge*, *old food*, *dish towel*, *hand held brush*, *vacuum*, *panel*, *health care reform*, ...), particularly at the size they are in the field of view. Ditto for verbs denoting

actions (e.g., *clean*, *speak*, *sit*, *stand*, *open*, *renovate install*, *bend*, *cook*, *mix*, ...). The system is not really grounding the meanings of these words in video. Rather it is just indexing based on surface features. This is what we mean when we say that our system uses *linguistics*. While this system may use techniques that are prevalent in the natural-language-processing community, and one might even call them *computational linguistics*, one would not call them *linguistics*. This is not to denigrate such a system. It is simply incomparable to our work.

5. Das et al. (2013) do not report measured alignment between the words in the text and the portions of the video. Thus one is unable to determine whether sentence generation really is based on video features that convey the meanings of the words and sentences generated or whether it is more based on accidental correlation with features in the background that are not reflective of the true meanings.

Rohrbach et al. (2013) generate text annotations for videos with a two-step process. They first translate a video x into an intermediate representation (SR) y and then translate the SR y into a sentence z. The SR is five discrete random variables (activity, tool, object, source, and target). There are 66 possible activities, 43 possible tools, 109 possible objects, 51 possible sources, and 35 possible targets. The mapping from video to SR is mediated by a joint probability model implemented as a conditional random field (CRF) that mutually constrains these five random variables. This CRF is trained in a supervised fashion. The training data contains videos paired with human annotated SRs. The mapping process from video to SR yields a quantized SR by returning the SR from the training set with the lowest Hamming (1950) distance to the SR estimated by the CRF.

The text-generation process involves a second step which maps an SR to a sentence. However, this process does not use any information from the video that is not already abstracted in the SR. For purposes of comparing with our work, this process is not relevant. The discrete quantized SR is the component of the work of Rohrbach et al. (2013) that is most analogous to the individual words that we generate. In our case, the mapping from words to sentences is done by a deterministic grammar and neither introduces errors nor contains any other joint-distribution information to filter out errors. Their mapping from SR to sentences can introduce errors but also constitutes an additional level of joint-constrained-distribution information that can filter out errors. Thus we compare our system to only the first step of their work.

Due to Hamming-distance post processing, Rohrbach et al. (2013) can output only one of 5,609 possible SRs out of a total of $66\times43\times109\times51\times35$=552,175,470 ones that are nominally possible. Thus it can only generate 5,609 possible sentences. In contrast, our system can generate 235,575 possible sentences with no more than three objects for the first corpus, 406,296 possible sentences with no more than three objects for the second corpus, 6,614,325 possible sentences with no more than four objects for the first corpus, and 13,633,272 possible sentences with no more than four objects for the second corpus. Thus while on the surface, it appears that the system of Rohrbach et al. (2013) handles unrestricted text, in reality it handles a space of sentences that is four to five orders of magnitude smaller than we do. Thus they are solving an immensely easier problem with a much smaller space of possible outputs. Yet, Rohrbach et al. (2013, Table 1) indicate that they obtain the correct SR only 21.6% of the time. We obtain a true sentence more than

64% of the time. Moreover, our system learns solely from videos paired with sentences. Their system requires additional human annotation of the SRs associated with each video.

Points 1–5 from our comparison with the work of Das et al. (2013) also apply to the work of Rohrbach et al. (2013). In particular, the vast majority of the object classes are well beyond the state-of-the-art ability to support recognition if it were not for the CRF (e.g., *avocado*, *egg*, *cucumber*, *bag of chilies*, *cutting board*, *loaf of bread*, *lime*, *knife*, *plate*, *butter*, *carrot*, *(half) kiwi*, *package of beans*, *orange*, *saucer*, ...), particularly at the size they are in the field of view. Similarly, the vast majority of the verbs are well beyond the state of the art to support in action recognition, if it were not for the CRF (e.g., *slice*, *crack*, *take out*, *rinses*, *put away*, *select*, *split*, ...). Rohrbach et al. (2013) derive most of their success from the highly constrained set of possible SRs and the distribution encoded in the CRF. That means it cannot describe videos that exhibit a person taking a kiwi out of the fridge if that never occurred in the training corpus, even though it might be a perfectly reasonable video. Surely vastly more than 5,609 of the 552,175,470 possible SRs are plausible and perhaps even likely. Yet even with this constraint, Rohrbach et al. (2013, Table 2) report that when human judges evaluated the truth of the generated sentences, the average report was 3.1 on a scale from 1 to 5, 3 being "70–80% good." Moreover, they are limited to the particular representation employed for SRs. They can only encode sentence meanings that are formulated in terms of the particular five random variables (activity, tool, object, source, and target). In contrast, our approach can formulate sentence meanings in terms of arbitrary conjunctions of any predicates applied to any subset of event participants so long as those predicates can be formulated as HMMs over arbitrary output distributions over features that can be extracted from the video.

Guadarrama et al. (2013) describe a method that outputs three-word sentences to summarize video activity. Like Rohrbach et al. (2013), such are encoded as three variables: an actor (subject), an action (verb), and an object (object). There are 45 possible subjects, 218 possible verbs, and 241 possible objects. Given a training corpus comprising video clips paired with annotated SVO triples, the method first builds three semantic hierarchies, represented as trees, one for each of subject, verb, and object, that indicate the similarity relationships among the meanings of the words that occur in the training corpus. Each word that appears in the training corpus constitutes a leaf node in one of the hierarchy trees. The internal nodes represent sets of dominated leaf nodes, a generalized concept having less specificity than the leaf nodes.

A visual classifier is associated with the leaf nodes for each individual subject, verb, and object. The leaf classifier uses

1. Dense Trajectories (Wang, Kläser, Schmid, & Liu, 2011, 2013; Wang & Schmid, 2013), encoded using a pre-trained codebook,
2. a vector of object-detector scores, each entry denoting the maximal score for each object class, and
3. a multi-channel approach that combines the above two features and classifies them with a non-linear SVM.

Once the classifiers are trained, probability estimates for the nodes in the hierarchy trees are obtained for an unseen video clip. Then nodes from the three hierarchies representing words to be generated for the unseen video clip are predicted by optimizing a cost function that trades off specificity for accuracy. When an internal node is predicted, the represen-

tative leaf word is selected from the set of leaves dominated by that node as the leaf that has the highest cumulative WUP in WordNet (Miller, 1995; Fellbaum, 1998). Guadarrama et al. (2013) also introduce a zero-shot approach to generate verbs that do not appear in the training corpus and thus are absent from the verb hierarchy. To do that, the verb is determined with text-mined likelihoods that fit the detected subject and object. While they paint this as a virtue, we view it as a deficit. Essentially, it is guessing the verb. While there are some celebrated cases where objects predict verbs and vice versa (e.g., *hammer*), we believe that this accounts for far less in actual video. When one sees a *dog* and a *cat*, there are still a plethora of possible verbs: *approach*, *leave*, *run away from*, *fight with*, *ignore*, *chase*, *flee from*,*bite*, *lick*, ... It is easy to pick examples that showcase where this works but that says little about how well the approach works in general.

The approach taken by Guadarrama et al. (2013) is very similar to that taken by Rohrbach et al. (2013), in that both construct joint probability models of a collection of random variables, five in the case of the latter but three in the case of the former. Rohrbach et al. add quantization by Hamming distance that is absent in the work of Guadarrama et al., and Guadarrama et al. add the zero-shot approach along with the hierarchies that balance between accuracy and specificity that is absent in the work of Rohrbach et al.. Without the above zero-shot extension, Guadarrama et al. output one of $45\times218\times241$=2,364,210 possible sentences. This number is roughly equivalent to the number of sentences that our method can produce.

The work of Guadarrama et al. exhibits the same shortcomings as in the work of Rohrbach et al. and Das et al. (2013). Points 1–5 from our comparison with the work of Das et al. also apply to the work of Guadarrama et al.. As is the case with the work of Rohrbach et al., the vast majority of the object classes are well beyond the state-of-the-art ability to support recognition (e.g., *chef*, *cook*, *microphone*, *flute*, *flour*, *music*, *pasta*, *spaghetti*, ...), particularly at the size they are in the field of view. Similarly, the vast majority of the verbs are well beyond the state of the art to support in action recognition (e.g., *slice*, *cut*, *chop*, *prepare*, *make*, ...). It is almost certain that the visual-feature space would not separate verbs such as *chop* and *cut*, and nouns such as *pasta* and *spaghetti*. These words are also so similar in their semantic meanings that it is even quite difficult for humans to distinguish in short video clips. Thus while the number of words that can appear in the generated sentences is increased by considering similar lexical items, the difficult of the generation task does not increase as much as expected if the evaluation is lax (e.g., considering *chop* to be correct even though *slice* actually happens in the video).

On the other hand, because of the specificity-accuracy tradeoff, the generated sentences sometimes are uninformative, e.g., *An animal plays something* and *An animal does something with the instrument* (Guadarrama et al., 2013, Table 4). Also the zero-shot approach seems to override the actual activity recognition quite easily, as can be seen in the fourth row (Guadarrama et al., 2013, Table 4): *A car rides the vehicle*. Finally, Guadarrama et al. do not evaluate the truth of the sentences generated. Instead, they only calculate the WUP similarity between generated and annotated subjects, verbs, and objects, independently. By our estimates, seven out of the eleven generated sentences in are false of the corresponding video clip. These are the examples picked to showcase their system. Presumably, it performs worse on other examples. It is unclear what the actual truth accuracy of the generated sentences is over the entire corpus.

There is also something deeply unsettling about the general approach taken by both Rohrbach et al. and Guadarrama et al. of using a joint probability model derived by text mining to influence activity recognition. Suppose that a corpus of text had much higher frequency of occurrence of *dog chases cat* than *dog is-bigger-than cat* or *dog eats-with cat*. That says nothing about the actual prior truth probability of the underlying propositions, let alone the actual posterior truth probability conditioned on a particular video. In some sense, Rohrbach et al. and Guadarrama et al. are actually not grounding language in video but rather generating natural-language utterances using information obtained from ungrounded language.

7. Discussion

The computational linguistics community has become accustomed to employing large lexicons and grammars trained on large text corpora to process unrestricted text. Similarly, the computer vision community has become accustomed to employing methods that can be trained on large image and video corpora to process unrestricted images and video. One may wonder what it would take to extend the methods explored here so that they too can apply to large-scale unrestricted text and video corpora. One might assume that it is simply a matter of employing better state-of-the-art methods from both computational linguistics and computer vision. This, however, is not the case. While we do not use state-of-the-art methods from computational linguistics, our computer vision methods are state of the art. We use the deformable part model (DPM) object detector (Felzenszwalb et al., 2010a, 2010b) and an action detector that exhibits state-of-the-art performance. Our approach is limited by computer vision, not computational linguistics. The state of the art in object detection is reflected by the ongoing Pascal Visual Object Category (VOC) Challenge (Everingham et al., 2010). It currently has 20 classes and current state-of-the-art performance is about 40-50% on the best classes, and far worse for other classes. The state of the art in action recognition is reflected by the standard corpora used in that community, e.g., Weizmann (9 classes; Blank, Gorelick, Shechtman, Irani, & Basri, 2005), KTH (6 classes; Schuldt, Laptev, & Caputo, 2004a), UCF Sports (10 classes; Rodriguez, Ahmed, & Shah, 2008), UCF YouTube (11 classes; Liu, Luo, & Shah, 2009), and Olympic Sports (16 classes; Niebles, Chen, & Fei-Fei, 2010). The best reported performance on these corpora (Weizmann 100%: Tian, Sukthankar, & Shah, 2013b; KTH 95.49%: Yuan, Li, Hu, Ling, & Maybank, 2013; UCF Sports 95%: Sadanand & Corso, 2012; UCF YouTube 89.4%: Zhu, Wang, Yang, Zhang, & Tu, 2013; and Olympic Sports 85%: Gaidon, Harchaoui, & Schmid, 2014) might lead one to the mistaken conclusion that action classification is solved for small numbers of classes. However, Barbu, Barrett, Chen, Siddharth, Xiong, Corso, Fellbaum, Hanson, Hanson, Hélie, Malaia, Pearlmutter, Siskind, Talavage, and Wilbur (2014) illustrate that this is false, a 6-class corpus for which state-of-the-art methods get no more than 52.34%. Moreover, the largest corpora actively used for action recognition contain about 50 classes (UCF50, Reddy & Shah, 2013 and HMDB51, Kuehne et al., 2011). The best reported performance on UCF50 is 91.2% and on HMDB51 is 57.2%, (Wang & Schmid, 2013). Thus an approach, such as ours, which grounds the meaning of each individual word in state-of-the-art computer vision object detectors, trackers, and action recognizers is inherently limited to a very small number of concepts. The space of natural-language utterances that

one can erect around such is thus limited and can effectively be captured by a small fixed unambiguous context-free grammar. Thus employing state-of-the-art methods from computational linguistics would not improve the generality of our approach given the limited state of the art in computer vision.

In this paper we, thus, do not employ such state-of-the-art methods from computational linguistics. We employ a small fixed lexicon and grammar. We make no claim that this lexicon and grammar is general. The particular lexicon and grammar is not the focus of this work. They serve to illustrate our framework and the capability of that framework for supporting the concerns outlined at the end of Section 1. One can change the lexicon or grammar and still use our framework. Indeed, we have done so and report some of this. Table 11(a) reports two slightly different grammars. The experiments employ two different video corpora with two different sets of sentential annotations that use these different grammars as reported in Tables 1–3. These video corpora use different sets of objects and associated object-detector models. Barbu, Siddharth, and Siskind (2014) employ yet another corpus, with a different set of objects and object-detector models, a different lexicon, a different grammar, and a different set of word-meaning representations. This demonstrates that our framework can be adapted to a variety of such. But beyond this, Barbu et al. (2014) demonstrate yet another whole different application of the same framework, namely video retrieval. And they do so on a corpus of ten full-length Hollywood movies. This corpus is far from "toy." Our framework can support such large-scale real-world video "in the wild." Yet the concept vocabulary is still small so the natural-language fragment is still restricted. While one could employ state-of-the-art methods from computational linguistics, the supported concept set and thus the supported language fragment would still be small. Thus one would not be using these state-of-the-art methods to the potential that they were designed for.

There are two general approaches towards action recognition in computer vision. One employs methods to detect and track people and objects that participate in the action, classifying action by properties derived from the detected objects and tracks. The other extracts and classifies features from video without detecting and tracking people and objects. The latter methods generally employ a bag of spatio-temporal visual-words approach (BOW). They generally extract feature vectors, such as spatio-temporal interest points (STIP; Schuldt, Laptev, & Caputo, 2004b), at a subset of space-time points, build a codebook by pooling such, vector quantize such feature vectors on this codebook, compute a histogram of codebook-entry occurrences on the pooled frames of a video, and classify these histograms with temporally invariant models. Early approaches to action recognition generally employed the former method (e.g., Siskind & Morris, 1996; Mann, Jepson, & Siskind, 1996, 1997; Siskind, 1999, 2000, 2001; Fern, Givan, & Siskind, 2002a; Fern, Siskind, & Givan, 2002c; Fern, Givan, & Siskind, 2002b; Siskind, 2003). This approach was eschewed in more recent work, in favor of the latter method, because of the difficulty of detecting and tracking people and objects reliably (e.g., Schuldt et al., 2004b; Liu et al., 2009; Ikizler-Cinbis & Sclaroff, 2010; Kuehne et al., 2011; Reddy & Shah, 2013). However, the BOW approach suffers from a severe limitation: it does not localize the event participants. While it may be able to generate verbs to describe classified actions, it cannot generate nouns to describe the object class of event participants, adjectives to describe the properties of event participants, spatial-relation prepositions to describe the relative position of event

participants, adverbs to describe event properties, or motion prepositions to describe the path taken by event participants. This is a distinguishing, novel, and unique aspect of our approach. Moreover, while some systems, such as the one proposed by Guadarrama et al. (2013), employ an object detector in addition to a STIP-based event detector, they do not link the objects as arguments to the event predicates. Any system using a similar approach like this would

1. fail to distinguish *the dog approached the person* from *the cat approached the person* when both a dog and a cat were present in the field of view and
2. fail to distinguish *the dog approached the person* from *the person approached the dog.*

Our approach correctly makes such distinctions. One of the design principles behind our corpus was that multiple people appear in most, if not all, videos, and most, if not all, objects appear in every video. Beyond this, most videos depict simultaneous different actions by different subsets of the participants. This is what renders the minimal-pairs experiment (Section 5.3) and the acquisition experiment (Section 5.5) far from trivial.

We make no claim that the particular features that we employ in Tables 6 and 8 are sufficient to represent the semantics of all possible words and utterances. These serve just to support the experimental evaluation conducted in Section 5. One could employ the same sentence-tracker approach discussed here with a different set of features. Indeed, we have done so (Barbu et al., 2014). Moreover, we make no claim that one can employ HMMs that form the core of the sentence tracker to represent the semantics of all possible words and utterances. This is not just a limitation of an HMM-based approach that requires object detectors and trackers; BOW approaches suffer from this as well. A BOW approach cannot represent the verb *approach.* And neither a BOW or HMM approach can represent the verbs *liberate*, *contemplate*, *discuss*, *help*, *finish*, ... Representing the semantics of the entire space of verbs, let alone all of natural language, even in a non-grounded fashion, and even more so, grounded in video, is the central unsolved problem of all of computational linguistics, AI, and cognitive science.

On the surface, it may appear that BOW approaches can be more robust at recognizing certain action classes like *play an instrument* than approaches that involve detecting and tracking objects. However, none of the standard datasets (Weizmann, KTH, UCF Sports, UCF YouTube, Olympic Sports, UCF50, or HMDB51) have a class *playing an instrument* (in general). Only one, UCF50, has classes for playing a small number of specific instruments: *drumming*, *playing guitar*, *playing piano*, and *playing violin.* We are unaware of any published action-recognition systems that perform well on this dataset. One of the best performing methods on this dataset is Action Bank (Sadanand & Corso, 2012), but it does not use BOW. The performance of this method is enlightening as to the current state of the art. It gets only roughly 80% accuracy on these classes. Moreover the confusion matrix is enlightening: *drumming* is confused with *biking* and *yoyo*, *playing piano* is confused with *basketball*, *drumming*, *golf swing*, *tennis swing*, *soccer*, and *juggling*, and *playing violin* is confused with *drumming*, *rope-climbing*, *taichi*, *tennis*, *yoyo*, and *rock climbing.* These confusions indicate that it lacks any deep understanding of the characteristic of the actions in question and appears to be triggering off of spurious correlations with the particular dataset. In particular, the dataset does not contain people sitting next to a drum set or piano, or holding a guitar or violin without playing it. So there is no way to know whether it is actually recognizing the *playing* activity or simply recognizing gross image statistics

that indicate that such instruments are present in the field of view. That is before one gets to motions such as *air guitar*, *banging the piano keys with your elbow*, or simply *waving a violin in the air* that constitute activity with the instrument in question but don't constitute playing said instrument. Beyond this, we see little ability for it to generalize from playing a specific instrument to playing an instrument in general.

BOW-based systems often do not encode the true semantics of the actions in question. They often trigger off of spurious correlations in the dataset. This has been acknowledged by authors of such systems themselves.

> "For instance, *v_spiking* normally happens in a crowd of people, and *diving* happens in a pool. This is common for professional sport actions which take place in highly structured environments (Liu et al., 2009, p. 2002)"

> "Basketball shooting and volleyball actions are also confused in some cases: this is largely because most of the time, the basketball and volleyball sports use very similar courts (Ikizler-Cinbis & Sclarof, 2010, p. 505)"

One may desire, or even expect, some form of characterization of the space of possible words or videos that our approach can support. Unfortunately, we know of no way to provide such. We know of no way, in general, of formally characterizing the space of words, images, or video that can be supported by any action-recognition system, or for that matter any object-recognition system or, more generally, any computer vision, computational linguistics, or AI system.

Our current corpus lacks camera motion. But this is not a restriction of our approach. This restriction does not appear in any of the mathematical or algorithmic formulations in Section 2 and 4, or even in the implementation. The sentence tracker is an extension of prior work on detection-based tracking (Barbu et al., 2012b) which was employed to perform action recognition and sentence generation on videos that do involve camera motion (Barbu et al., 2012a). Barbu et al. (2014) apply the sentence tracker to perform video retrieval on a corpus of ten full-length Hollywood movies, the vast majority of which involve camera motion.

Our framework is expressly not restricted to using only verbs to represent events. Our current linking process and the particular grammar used to support that process is restricted to such. But nothing turns on that. As discussed above, the sentence tracker can use *any* linking process to construct any factorial utterance-level HMM out of constituent word-level HMMs. For expedience, we limit the set of features entertained during learning on a part-of-speech basis. This restriction could be lifted with no change to the algorithm or its implementation. It was introduced to allow convergence with a smaller training set. We know of no reason why the method from Section 4 would not work without such a restriction. It would require a larger corpus that would be unwieldy to perform experiments with.

Our method represents word meanings in all parts of speech simply as predicates over one or more tracks and sentential meanings as conjunctions of such. Presumably, a different linking process could construct the same logical form $\textsc{man}(x) \wedge \textsc{pause}(\textsc{x})$ from both sentences like *The man made a pause* as well as it could from *The man paused*. This is the beauty of our approach, employing a unified representation for the meanings of all

words in all parts of speech, a common cost function, a common algorithm, and a common implementation.

State-of-the-art object-recognition systems are highly unreliable. For most image datasets, a trained object model, for say *person* or *chair*, may succeed on one image and fail on another, even if it is of the same chair or same person in the same pose wearing the same clothing in the same background. For most video, this even happens between adjacent frames of the same video. State-of-the-art object detection suffers from immense false positives and negatives. Moreover, not only does reliable object detection not imply reliable action recognition, state-of-the-art action recognizers are similarly highly unreliable. State-of-the-art recognizers for *bend* and *wave* trained on one dataset yield chance performance on a different datasets. Even on the same dataset, action recognizers can mysteriously both succeed and fail on very similar samples, with the same background, same actors, same manner of performance of action, *etc.* The central novel contribution of this work is the sentence tracker in Equation 10, a method for overcoming the severe limitations of both object detectors and action detectors by formulating a joint model of object detection, tracking (temporal coherence), and sentential semantics.

While our video corpora may appear to be simpler than those typically used for current action-recognition work in the computer vision community (e.g., Weizmann, KTH, UCF Sports, UCF YouTube, Olympic Sports, UCF50, or HMDB51) this apparent simplicity is misleading. Several aspects of our video corpora are far more complex than those used in the vast majority of related work.

1. Most videos contain many, if not all, of the objects in our repertoire. This makes language acquisition difficult. One needs to determine which objects are being referred to by the training sentences and ignore the extraneous ones in the field of view. This is all done automatically without any human annotation.
2. Most videos contain at least two simultaneous actions, often performed by different people on different objects. One needs to determine which action is being referred to by the training sentence associated with that video, pay attention to the particular subset of people and objects that participate in that action, and ignore the extraneous activity that occurs in the field of view. This is all done automatically without any human annotation.
3. Our system can process complex natural-language sentences that contain many participants, e.g., something as complex as *The person to the left of the chair carried the backpack to the right of the traffic cone towards the stool to the left of the person.* It can even support multiple instances of the same noun in a sentence that refer to distinct instances of that object class in the video (as in *person* above). It can determine the semantic-role assignment, which nouns and which arguments of which words correspond to which regions in the video frames. Such assignment is determined automatically without any human annotation and can change with small and subtle changes to the sentence. Moreover, we can learn solely from such complex sentential annotation, without any human annotation of which words correspond to which regions in the video frames.

Our novel and central technical contribution is the formulation of the sentence tracker in Equation 10 and the observation that it can be optimized using standard well-known techniques adapted from HMMs, namely the Viterbi algorithm (1967) and Baum-Welch

(1970, 1972). The key to understanding Equation 10 is that it jointly optimizes a cost function that incorporates multiple detection-based trackers, one for each event participant, and multiple factorial event models, one for each lexical item in a sentence, judiciously linking the detection-based trackers to the factors of the sentential model in a way consistent with the predicate-argument structure of a sentence, to model the truth-conditional semantics of a sentence and how it is derived from its constituent words. Formulating truth-conditional sentential semantics in this way allows exiting algorithms like Viterbi and Baum-Welch to ground the semantics of natural language in video and perform novel applications such as language inference (Section 5.3), language generation (Section 5.4, and language acquisition (Section 5.5), particularly the minimal-pair experiment in Figure 15 and acquisition from videos labeled with whole sentences and no further human annotation.

There has been significant prior work on multi-object tracking (e.g., Berclaz, Fleuret, Turetken, & Fua, 2011; Pirsiavash, Ramanan, & Fowlkes, 2011). A novel aspect of the event tracker is that the particular formulation of detection-based tracking as a cost function that can be optimized by the Viterbi algorithm allows forming a joint model with an HMM-based event detector that can also be optimized by the Viterbi algorithm with a cross-product lattice. This might not be possible with other trackers and other event models. Beyond this, the sentence tracker forms a joint model of multiple trackers and a factorial HMM, linking particular factors to particular trackers, in a way that can again, also be optimized by the Viterbi algorithm with a cross-product lattice. This also might not be possible with other multi-object trackers and other event models.

Our video corpora were filmed by giving actors instructions about what actions to perform. As such, they were 'staged.' The computational linguistics community has attempted to use unsolicited samples of natural language for fear that solicited samples might introduce bias. One might wonder whether it is desirable, and even possible, to do so for video corpora as well. However, it appears infeasible to gather unsolicited video corpora except in surveillance situations. Surveillance video tends to be highly uniform and sparse: only a few event classes occur and most occur very infrequently. This renders it ill suited to action recognition. Almost all other situations where video is recorded, even when not recorded explicitly for computer vision use, is solicited. Most amateur video of the form uploaded to YouTube is similarly staged at some level as it usually records activity elicited specifically for filming. Indeed, most prominent video corpora used in the computer vision community to evaluate action recognition were filmed specifically for the purpose of constructing the corpus: Weizmann, KTH, the Activities of Daily Living corpus (Messing, Pal, & Kautz, 2009), the DARPA Mind's Eye corpus (both year 1 and year 2), and the TaCOS corpus used by Rohrbach et al. (2013), just to name a few. While the YouCook corpus used by Das et al. (2013) was culled from YouTube, the videos themselves appear to be staged, just as all of the above.

Some related work on generating sentences that describe video evaluates the generated sentences by comparison with human-elicited sentences for the same video. Such is often done by computing BLEU scores (Rohrbach et al., 2013) or measuring the fraction of words in common between the machine-generated and human-elicited descriptions (Khan, Zhang, & Gotoh, 2011). While such might evaluate the degree to which machine-generated sentences are natural sounding, it fails to evaluate the truth of the machine-generated sentences, the central objective of our work. Indeed, machine-generated sentences with

high BLEU scores or high commonality with human-elicited descriptions are often false of the video even when the human-elicited descriptions are true.

Our current linking process would fail with an ambiguous sentence parse. The linking process might also fail to yield an unambiguous role assignment and unique linking function. Further, our current lexicon contains no lexical ambiguity and our current linking process would not support such. Any of the myriad approaches to parsing and constructing logical form in the presence of ambiguity could be brought to bear on this problem. But beyond this, the current approach offers a novel possibility that no existing approach can support. One can imagine using video to disambiguate parsing and the construction of logical form. One could imagine evaluating the truth of various word senses, sentence fragments, attachment alternatives, and alternate logical forms against video using the sentence tracker.

The sentence tracker is a general-purpose inference mechanism for combining information from multiple frames of a video using both language and vision. While we have presented a particular instantiation of the sentence tracker, with particular detectors, particular temporal-coherence scores, and particular event models operating in 2D, the general approach could be instantiated in numerous other ways. We have employed object detectors as detection sources, but any method that selects image regions could be used in the approach presented. These need not be rectangular: one can imagine variants of the sentence tracker that employ general-purpose foreground-background segmentation instead of object detection. They also need not be two-dimensional: one can imagine variants of the sentence tracker that employ projection models to reconstruct temporally-coherent tracks in 3D from 2D images that also satisfy 3D event models. We could even pool the detections from a variety of sources and scale their scores to prefer more reliable ones when possible. Moreover, our temporal-coherence score uses only optical flow, but it could employ an appearance model in order to alleviate situations where tracks converge to the same image location and are swapped between the two tracked objects as they again diverge from that location. If one were to employ a human-pose detector, one could incorporate coherence of human-pose variation into the temporal-coherence model. One could similarly incorporate changing human pose into the event model. Doing so with the event tracker would allow such an event model to influence and improve the recovered human pose estimated in a top-down fashion, much in the same way that the event model can influence and improve the recovered tracks. Finally, while our event models are formulated as HMMs, more general frameworks are possible. Even nongenerative frameworks, like maximum-entropy Markov models, could be accommodated as long as inference could be performed using a lattice and dynamic programming. One can even imagine forgoing the lattice and dynamic programming to integrate more complex models of object detection, temporal coherence, and events using message-passing inference.

The sentence tracker can also learn word meanings from video paired with sentences. Unlike prior work, our method deals with video labeled with whole sentences, instead of individual words. Moreover, our method successfully learns without any prior delineation of the correspondence between words in the sentence labels to visual features in the associated video used for object and/or event recognition. The experiments show that it can correctly learn the meaning representations in terms of HMM parameters for our lexical entries, from highly ambiguous training data, where each training video clip depicts more than one

sentence and each sentence describes more than one clip. It does so by performing both inter- and intra-sentential inference: determining the meaning of a word cross-situationally from the collection of training samples in which it appears and well as by spreading the sentential meaning across the words in that sentence in a way that is consistent across a training set.

Our method is amenable to further extension. First, due to the nature of Markov models, each state depends only on its immediate predecessor. As discussed in Section 2.2, this property implies that the output model can only employ features computed on single frames or two adjacent frames. Such features may prove inadequate for larger lexicons. For example, our models often exhibit difficulty in differentiating between *picked up* and *put down*, since the only difference encoded is in the second-argument velocity orientation in the second state. Our current implementation computes this orientation using optical flow which can be noisy. One could more reliably differentiate between these two event classes if one could encode in the model the overall displacement of the second argument, along with the direction of that displacement, as the event proceeds: *picked up* involves a significant upward displacement while *put down* involves a significant downward displacement While it is not possible to encode such a multiple-frame feature in an HMM, it is possible to do so in more complex graphical models such as conditional random fields (CRFs). One can imagine employing CRFs as the event model, together with object detection and temporal coherence, in a variant of the sentence tracker.

Another possible extension is employing state-duration models in the HMMs. Without explicit state-duration models, the implicit state-duration model is exponential: the probability of staying in a state k for t frames is $a(k,k)^t$. While such an exponential state-duration model can encode a minimum duration for an event, to filter out short-term noise in the signal, as discussed in Section 2.2, it cannot bias an event detector towards a typical duration for performing an event. In our experiments, this can manifest itself in difficulty in distinguishing between *picked up* and *put down* because they have similar initial and final states but differ only in a short transition period. Employing explicit state-duration models, such as hidden semi-Markov models (HSMMs; Yu, 2010) as the event models within the sentence-tracker framework could potentially improve alleviate this difficulty.

A third possible extension is to employ discriminative training instead of maximum-likelihood training. Maximum-likelihood training makes use of only positive sentential labels on training data. Discriminative training can also make use of negative sentential labels. This could reduce the amount of training data required and also could yield better results as it trains the models competitively. Doing so would require a method for obtaining negative sentence labels. One could do so with manual annotation, just as for positive sentence labels. However, discriminative training works well when the number of negative labels far exceeds the number of positive ones. Thus rather than manual annotation, one can imagine some form of sentential inference to automatically generate negative sentential labels that could not possibly be true for a video with an associated positive sentential label. This may allow learning larger lexicons from more complex video without excessive training data.

8. Conclusion

We have presented a novel framework that utilizes the compositional structure of events and the compositional structure of language to drive a semantically meaningful and targeted approach towards event recognition. This multimodal framework integrates low-level visual components, such as object detectors, with high-level semantic information, in the form of sentential descriptions in natural language. Such integration is facilitated by the shared structure of detection-based tracking, which encodes the low-level visual features, and of the event models, in the form of HMMs, which encode the sentential semantics.

We demonstrated the utility and expressiveness of our framework by performing three separate tasks on our video corpora, simply by leveraging our framework in different manners. The first, **language inference**, showcases the ability to focus the attention of a tracker on the event described by a sentence, demonstrating the capability to correctly identify such subtle distinctions as between *The person picked up the chair to the left of the trash can* and *The person picked up the chair to the right of the trash can*. The second, **language generation**, showcases the ability to produce a complex sentential description of a video clip, involving multiple parts of speech, by performing an efficient search for the best description though the space of all possible descriptions. The third, **language acquisition**, showcases the ability learn a lexicon from a corpus of video clips annotated with sentential descriptions by searching among all possible lexicons to find one that allows the sentences to best collectively describe their associated video clips.

Acknowledgments

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0060. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

Appendix A. Our Linking Process

We use a linking process that is mediated by a grammar and portions of the lexicon Λ. The lexicon portion specifies the arity I and permissible roles of individual lexical entries. The grammar used for the experiments in Section 5 is shown in Table 11(a). The portion of the lexicon that specifies arity and permissible roles used in those experiments is shown in Table 11(b). With this grammar and lexicon portion, the linking process to be described below can determine that the sentence in Equation 11 has 3 participants and can produce the linking function in Equation 12.

The linking process Θ operates by first constructing a parse tree of the sentence $\mathbf{s}$ given the grammar. We do so by means of a recursive-descent parser. The lexical-category heads in this parse tree map to words used by the sentence tracker. Nominally, the lexical categories, e.g., noun (N), adjective (A), verb (V), adverb (Adv), and preposition (P), serve as *heads* of the corresponding phrasal categories NP, AP, VP, AdvP, and PP. The

(a)
$$
\begin{aligned}
\text{S} &\rightarrow \text{NP VP} \\
\text{NP} &\rightarrow \text{D [A] N [PP]} \\
\text{D} &\rightarrow \textit{an} \mid \textit{the} \\
\text{A} &\rightarrow \textit{blue} \mid \textit{red} \\
\text{N} &\rightarrow \textit{person} \mid \textit{backpack} \mid \textit{chair} \mid \textit{trash can} \mid \textit{traffic cone} \mid \textit{stool} \mid \textit{object} \\
\text{PP} &\rightarrow \text{P NP} \\
\text{P} &\rightarrow \textit{to the left of} \mid \textit{to the right of} \\
\text{VP} &\rightarrow \text{V NP [Adv] [PP}_\text{M}\text{]} \\
\text{V} &\rightarrow \textit{approached} \mid \textit{carried} \mid \textit{picked up} \mid \textit{put down} \\
\text{Adv} &\rightarrow \textit{quickly} \mid \textit{slowly} \\
\text{PP}_\text{M} &\rightarrow \text{P}_\text{M}\text{ NP} \\
\text{P}_\text{M} &\rightarrow \textit{towards} \mid \textit{away from}
\end{aligned}
$$

(b)
$$
\begin{aligned}
\textit{to the left of}&: \{\text{agent}, \text{patient}, \text{source}, \text{goal}, \text{referent}\}, \{\text{referent}\} \\
\textit{to the right of}&: \{\text{agent}, \text{patient}, \text{source}, \text{goal}, \text{referent}\}, \{\text{referent}\} \\
\textit{approached}&: \{\text{agent}\}, \{\text{goal}\} \\
\textit{carried}&: \{\text{agent}\}, \{\text{patient}\} \\
\textit{picked up}&: \{\text{agent}\}, \{\text{patient}\} \\
\textit{put down}&: \{\text{agent}\}, \{\text{patient}\} \\
\textit{towards}&: \{\text{agent}, \text{patient}\}, \{\text{goal}\} \\
\textit{away from}&: \{\text{agent}, \text{patient}\}, \{\text{source}\} \\
\text{other}&: \{\text{agent}, \text{patient}, \text{source}, \text{goal}, \text{referent}\}
\end{aligned}
$$

Table 11: (a) The grammar used for the experiments in Section 5. Terminals and nonterminals in red are used only for the experiments in Sections 5.3 and 5.4 on the first corpus. Terminals and nonterminals in green are used only for the experiments in Section 5.5 on the second corpus. Terminals and nonterminals in black are used for all experiments on all corpora. The first corpus uses 19 lexical entries over 7 parts of speech (2 determiners, 2 adjectives, 5 nouns, 2 spatial-relation prepositions, 4 verbs, 2 adverbs, and 2 motion prepositions). The second corpus uses 17 lexical entries over 6 parts of speech (1 determiner, 6 nouns, 2 spatial-relation prepositions, 4 verbs, 2 adverbs, and 2 motion prepositions). Note that the grammar allows for infinite recursion in the noun phrase. (b) The portion of the lexicon that specifies arity and permissible roles for the experiments in Section 5.

structure of the parse tree encodes the linking function between different words in the form of *government* relations (Chomsky, 1982; Aoun & Sportiche, 1983; Haegeman, 1992; Chomsky, 2002). This government relation can be defined formally as in Figure 19. For example, we determine that in Figure 20, the N *person* governs the P *to the right of* but not the N *chair*, and that the P *to the right of* governs the N *chair*.

The government relation, coupled with the lexicon portion, determines the number L of participants and the linking function θ. We construct a word w for each head. The lexicon portion specifies the arity of each lexical entry, namely the fact that *person*, *chair*,

- The lexical categories N, A, V, Adv, and P are *heads*.
- Parse-tree nodes α labeled with heads are *governors*.
- A parse-tree node α *dominates* a parse-tree node β iff β is a subtree of α.
- From X-bar theory (Jackendoff, 1977), a parse-tree node β is the *maximal projection* of a parse-tree node α iff
  α is labeled with a lexical category X,
  β is labeled with the corresponding phrasal category XP,
  β dominates α, and
  no other parse-tree node γ exists where
  γ is labeled with XP,
  β dominates γ, and
  γ dominates α.
- A parse-tree node α *m-commands* a parse-tree node β iff α and β do not dominate each other and the maximal projection of α dominates β.
- A parse-tree node α *c-commands* a parse-tree node β iff α and β do not dominate each other and α's immediate parent dominates β.
- A parse-tree node α *governs* a parse-tree node β iff
  α is a governor,
  α m-commands β, and
  no other parse-tree node γ exists where
  γ is a governor,
  γ m-commands β,
  γ c-commands β, and
  γ does not c-command α.

Figure 19: The government relation underlying the linking process Θ.

and *backpack* are unary and *to the right of* and *picked up* are binary. The sole argument for the word associated with each head noun is filled with a distinct participant.[3] The sole argument of the word associated with each unary non-noun head α is filled with the sole argument of the word associated with the head noun that governs α. The first argument of the word associated with each binary non-noun head α is also filled with the sole argument of the word associated with the head noun that governs α. The second argument of the word associated with each binary non-noun head α is filled with the sole argument of the word associated with the head noun that is governed by α. In the example in Figure 20, the sole arguments of the words associated with the nouns *person*, *chair*, and *backpack* are assigned the distinct participants 1, 2, and 3 respectively. The arguments of the word associated with the preposition *to the right of* are assigned to participants 1 and 2, since the N *person* governs the P *to the right of* which in turn governs the N *chair*. Similarly, the arguments of the word associated with the verb *picked up* are assigned to participants 1 and 3, since the N *person* governs the V *picked up* which in turn governs the N *backpack*.

3. This document does not concern itself with anaphora, thus we omit discussion of how to support potential coreference. Our implementation, in fact, does support such and mediates such by analysis of the determiners.

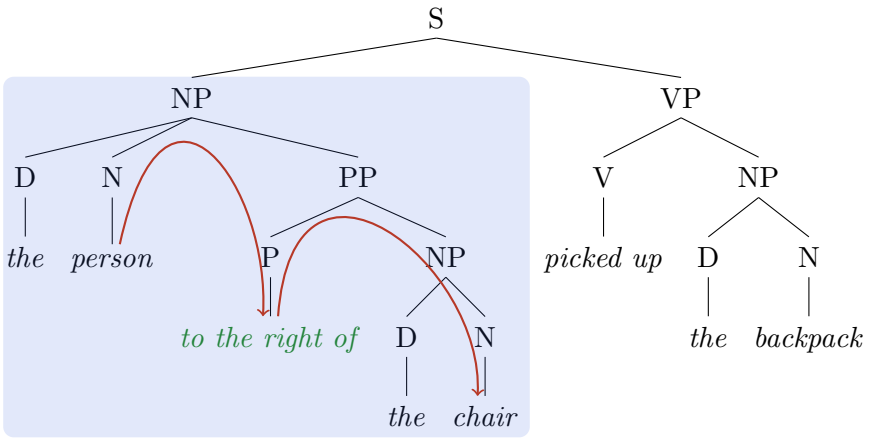


Figure 20: A parse tree for the example sentence *The person to the right of the chair picked up the backpack*. The highlighted portion indicates the government relations for the P *to the right of* that are used to determine its arguments. The N *person* governs the P *to the right of*, but not the N *chair*, and the P *to the right of* governs the N *chair*.

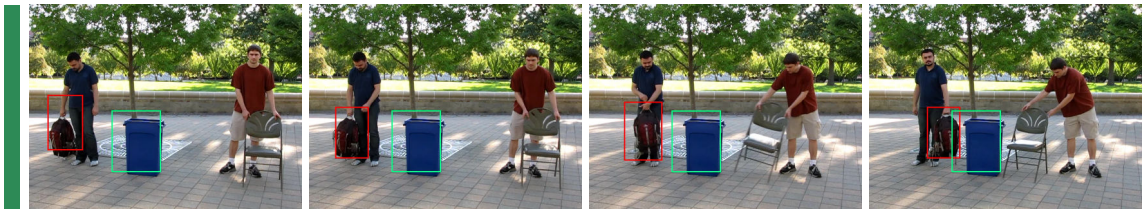
We further determine a consistent assignment of roles, one of agent, patient, source, goal, and referent, to participants. The allowed roles for each argument of each word are specified in the lexicon portion. A specification of the arity and permissible roles used for the experiments in Section 5 is given in Table 11(b). The specification $e: \{r_1^1, \ldots\}, \ldots, \{r_1^{I_e}, \ldots\}$ means that the arity for lexical entry e is I_e and $r_1^i, \ldots$ constitute the permissible roles for argument i. Each participant is constrained to be assigned a role in the intersection of the sets of permissible roles for each argument of each word where that participant appears. We further constrain the role assignment to assign each role to at most one participant. For the example sentence in Equation 11, the role assignment is computed as follows:

$$\text{role}(1) \in \{\text{agent}, \text{patient}, \text{source}, \text{goal}, \text{referent}\} \cap \{\text{agent}, \text{patient}\} \cap \{\text{agent}\}$$
$$\text{role}(2) \in \{\text{agent}, \text{patient}, \text{source}, \text{goal}, \text{referent}\} \cap \{\text{referent}\}$$
$$\text{role}(3) \in \{\text{agent}, \text{patient}, \text{source}, \text{goal}, \text{referent}\} \cap \{\text{patient}\}$$

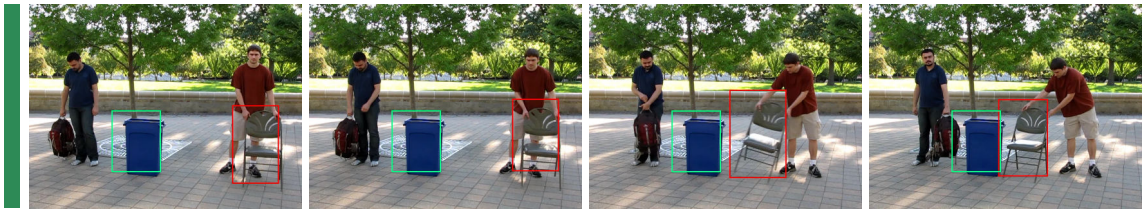
leading to:

$$\text{role}(1) = \text{agent} \quad \text{role}(2) = \text{referent} \quad \text{role}(3) = \text{patient}$$

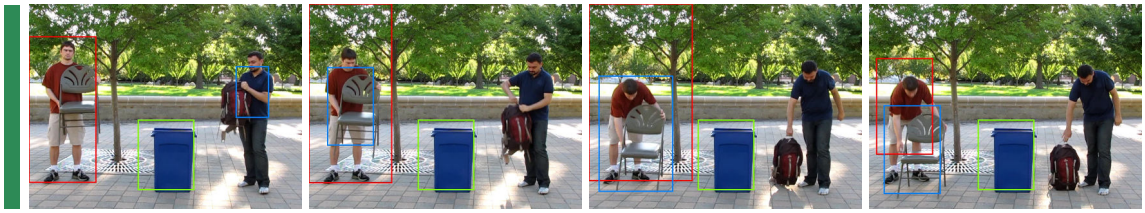
Appendix B. Complete Experimental Results



The backpack approached the trash can.



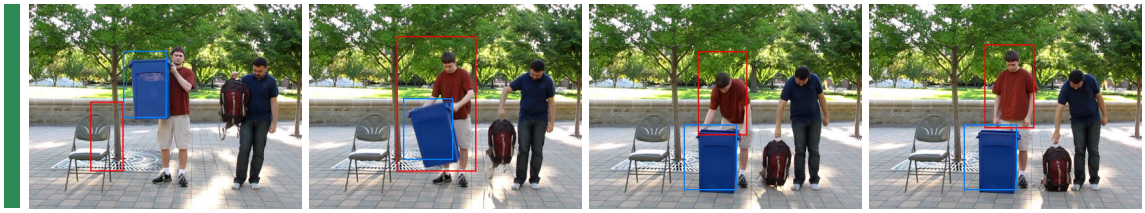
The chair approached the trash can.



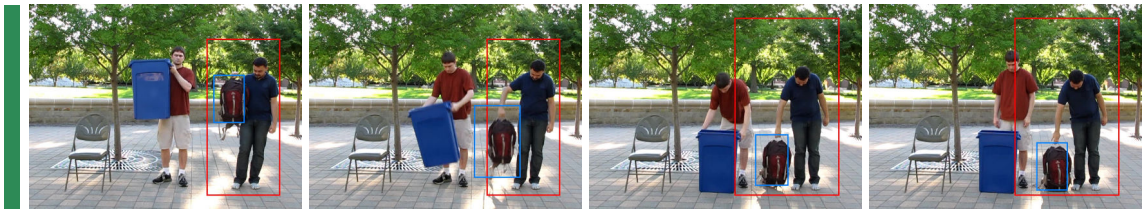
The person to the left of the trash can put down an object.

The person to the right of the trash can put down an object.

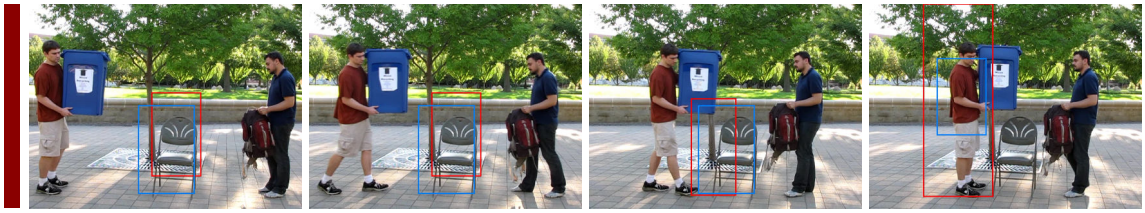
Figure 21: Language inference: two different track collections for the same video clip produced under guidance of two different sentences. The minimal pairs of sentences correspond to sentences 1–9 from Table 1 with the differences between the (a) and (b) variants highlighted. The track collections deemed by human judges to depict the given sentences are indicated in green, while ones that do not are indicated in red.



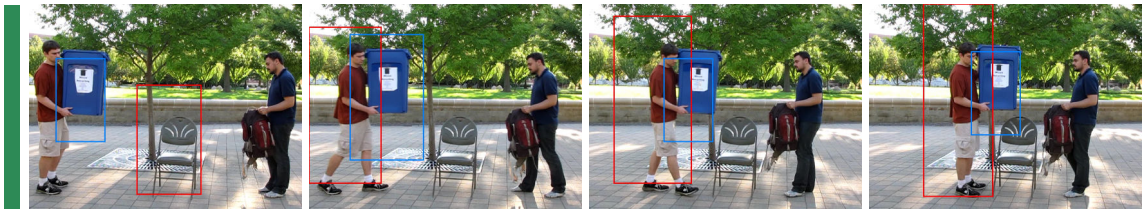
The person put down the trash can.



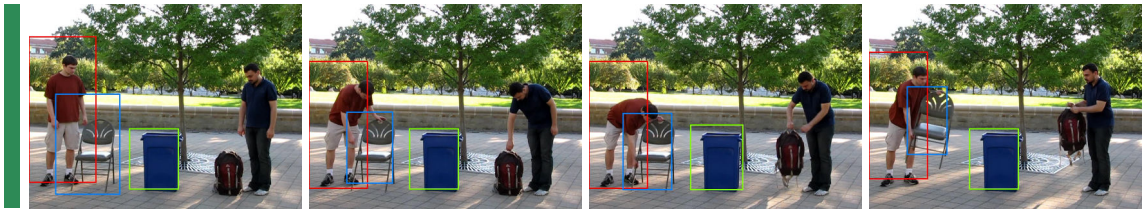
The person put down the backpack.



The person carried the red object.



The person carried the blue object.

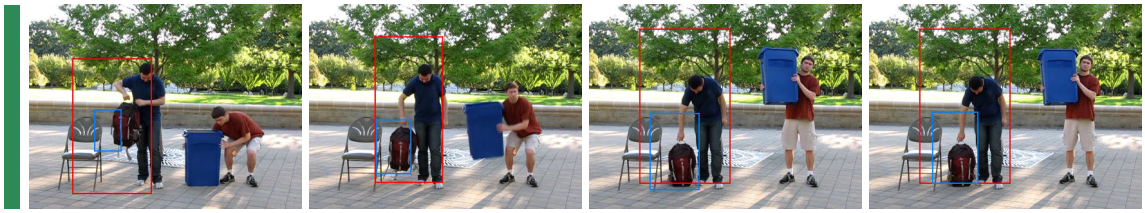


The person picked up an object to the left of the trash can.

The person picked up an object to the right of the trash can.

Figure 21: Language-inference examples continued.

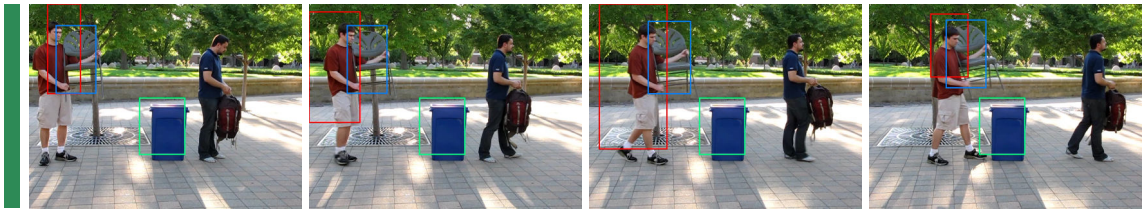
The person picked up an object.



The person put down an object.

The person picked up an object quickly.

The person picked up an object slowly.



The person carried an object towards the trash can.

The person carried an object away from the trash can.

Figure 21: Language-inference examples continued.

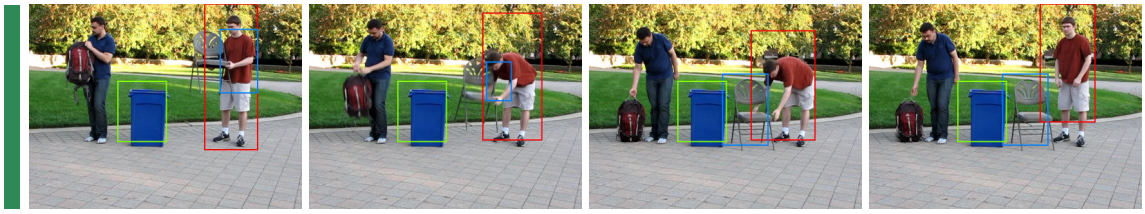
The backpack approached the trash can.

The chair approached the trash can.

The red object approached the chair.

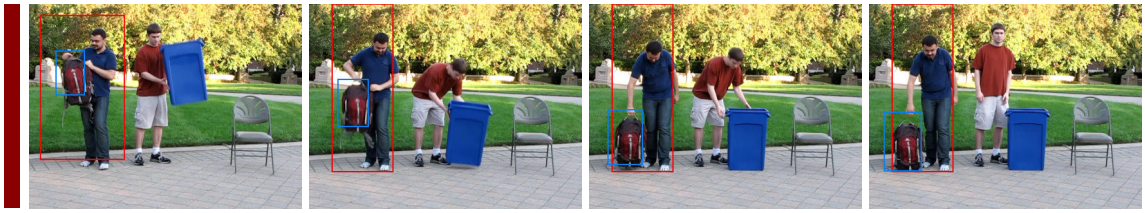
The blue object approached the chair.

The person to the left of the trash can put down an object.

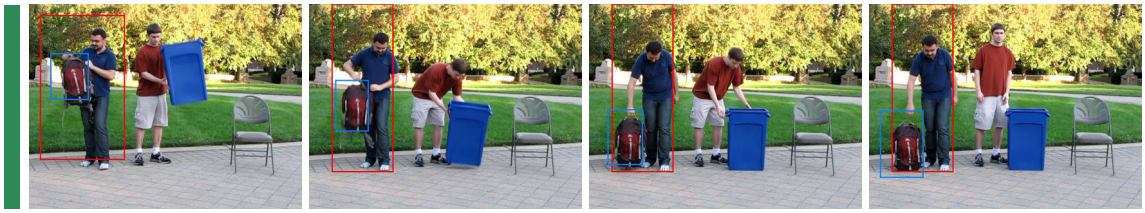


The person to the right of the trash can put down an object.

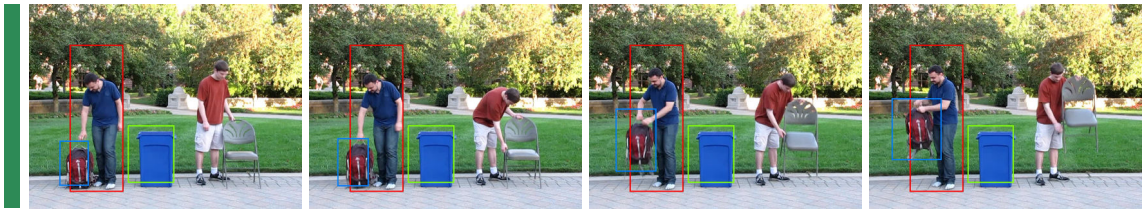
Figure 21: Language-inference examples continued.



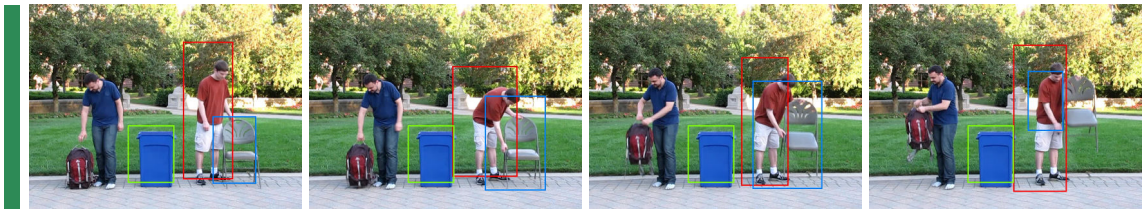
The person put down the trash can.



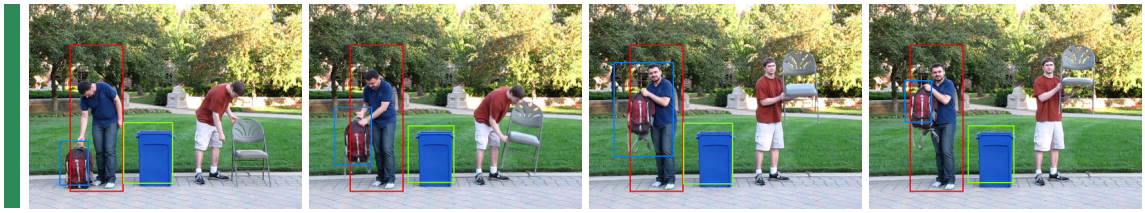
The person put down the backpack.



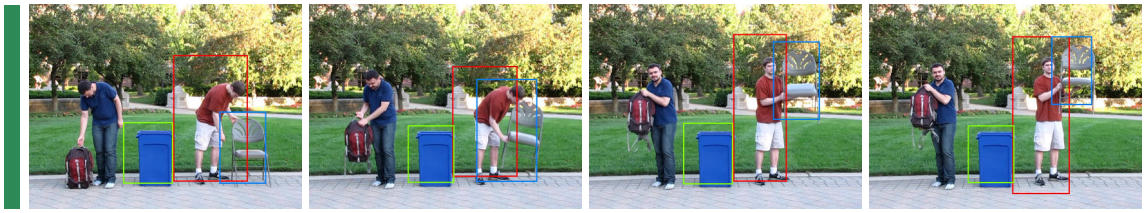
The person picked up an object to the left of the trash can.



The person picked up an object to the right of the trash can.

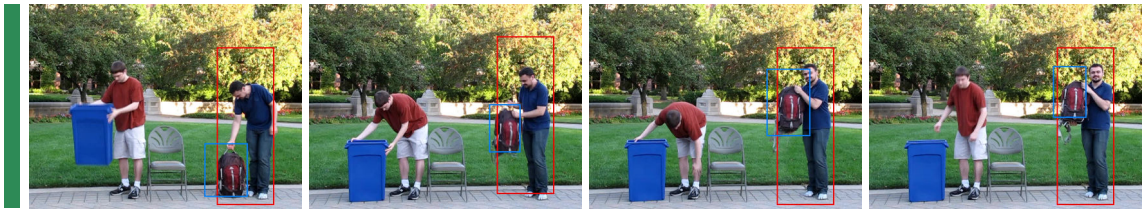


The person picked up an object to the left of the trash can.

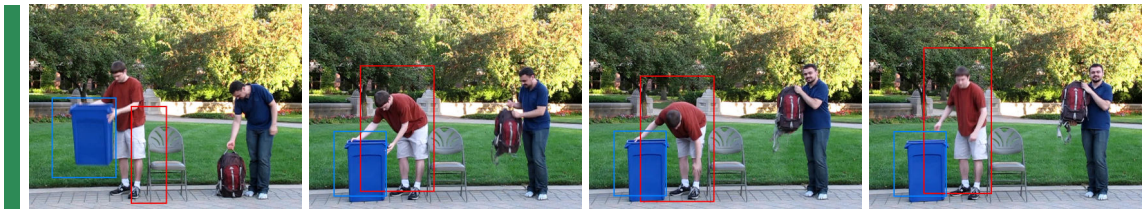


The person picked up an object to the right of the trash can.

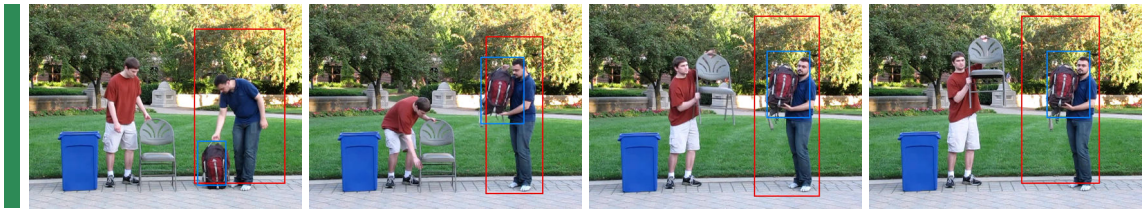
Figure 21: Language-inference examples continued.



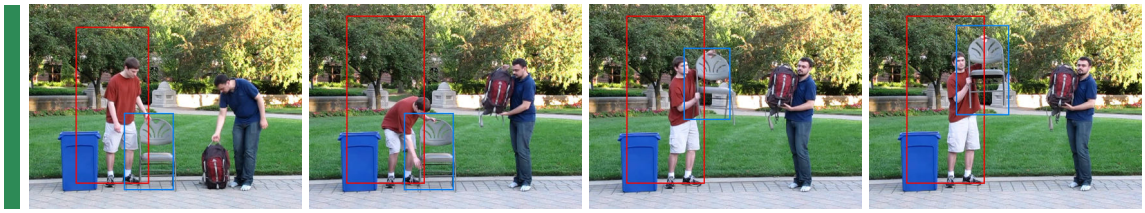
The person picked up an object.



The person put down an object.



The person picked up an object quickly.

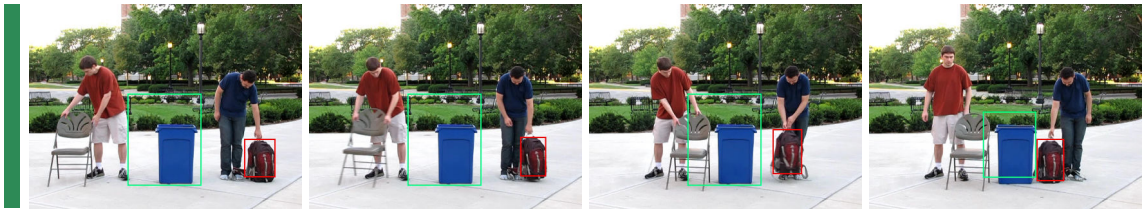


The person picked up an object slowly.

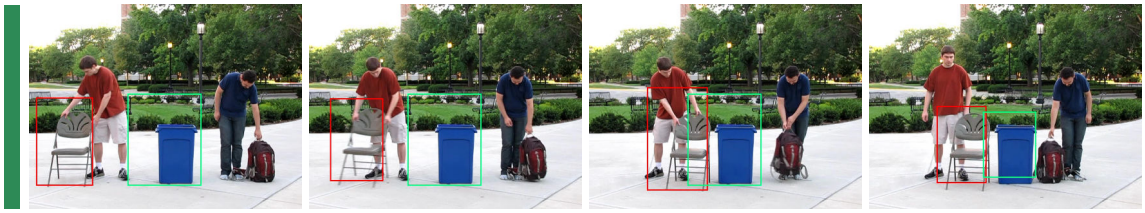
The person carried an object towards the trash can.

The person carried an object away from the trash can.

Figure 21: Language-inference examples continued.

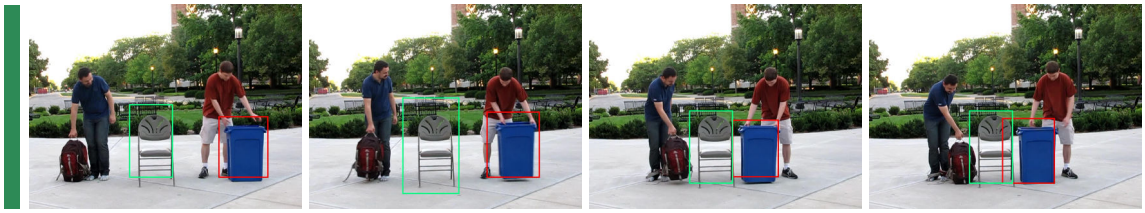


The backpack approached the trash can.



The chair approached the trash can.

The red object approached the chair.

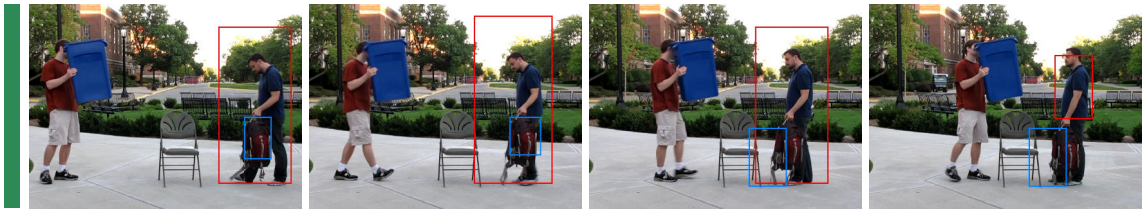


The blue object approached the chair.

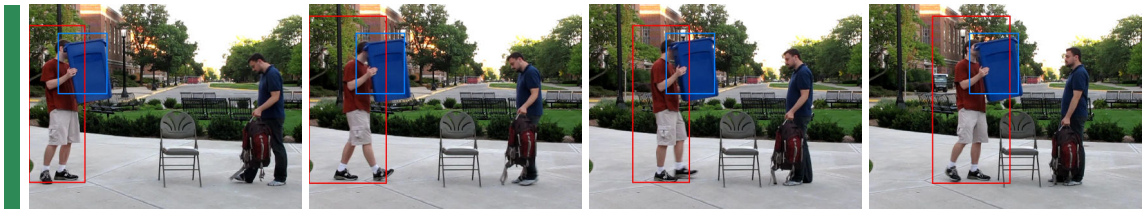
The person put down the chair.

The person put down the backpack.

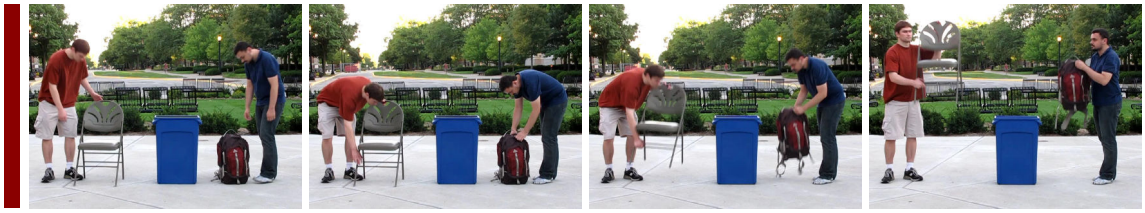
Figure 21: Language-inference examples continued.



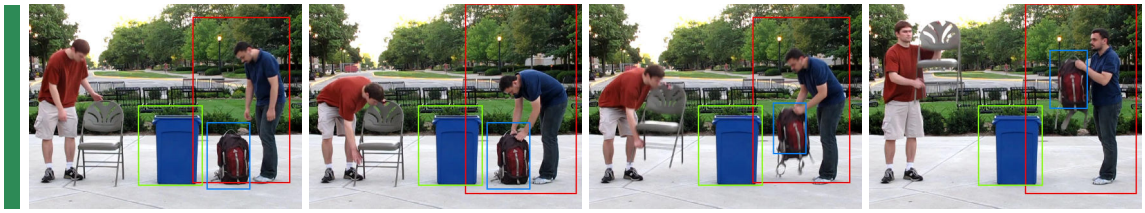
The person carried the red object.



The person carried the blue object.



The person picked up an object to the left of the trash can.



The person picked up an object to the right of the trash can.

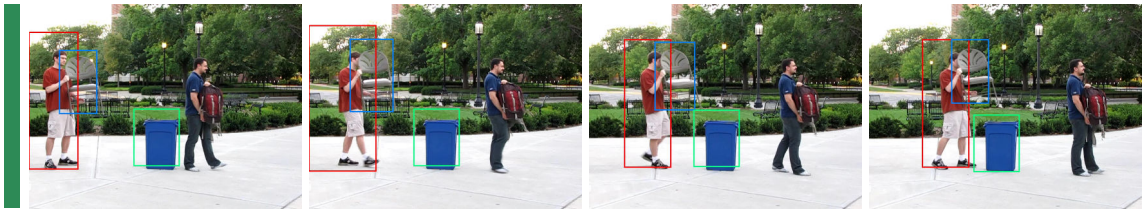
The person picked up an object.

The person put down an object.

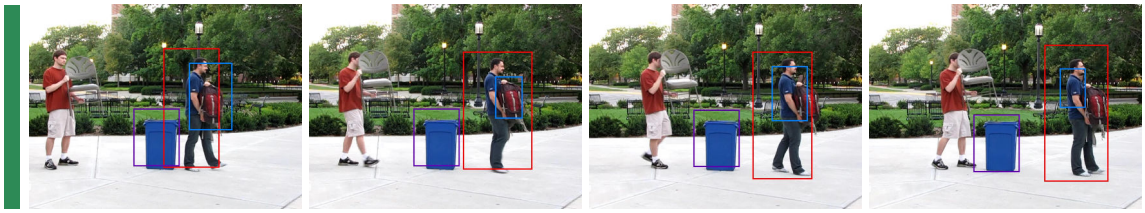
Figure 21: Language-inference examples continued.

The person picked up an object quickly.

The person picked up an object slowly.

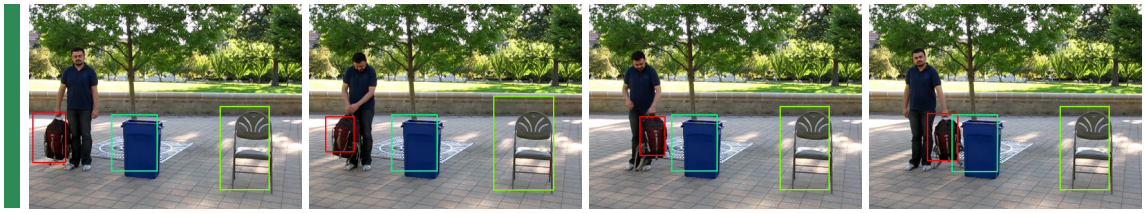


The person carried an object towards the trash can.

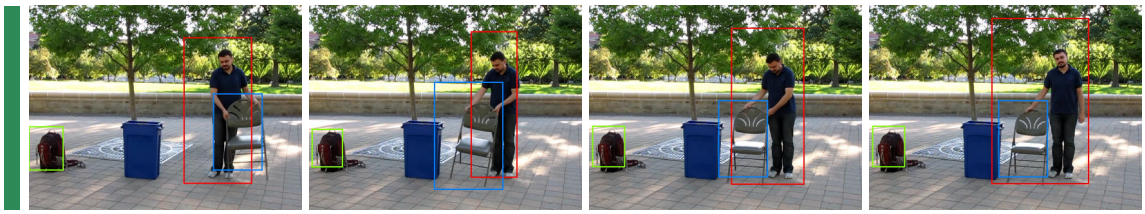


The person carried an object away from the trash can.

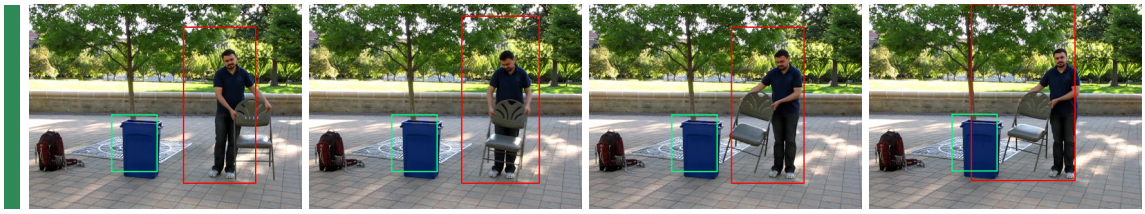
Figure 21: Language-inference examples continued.



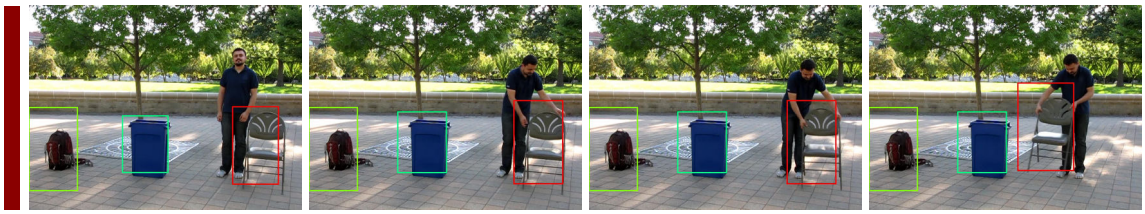
The backpack to the left of the chair approached the trash can.



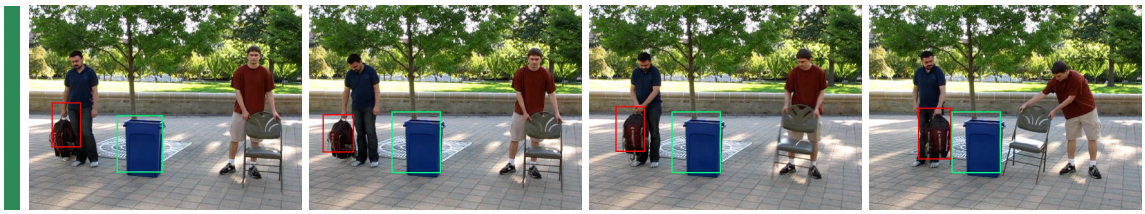
The person to the right of the backpack carried the chair.



The person to the right of the trash can approached the trash can.



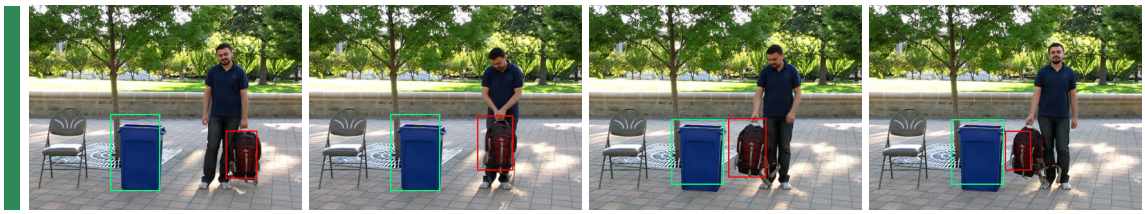
The chair to the right of the person approached the trash can.



The backpack to the left of the trash can approached the trash can.

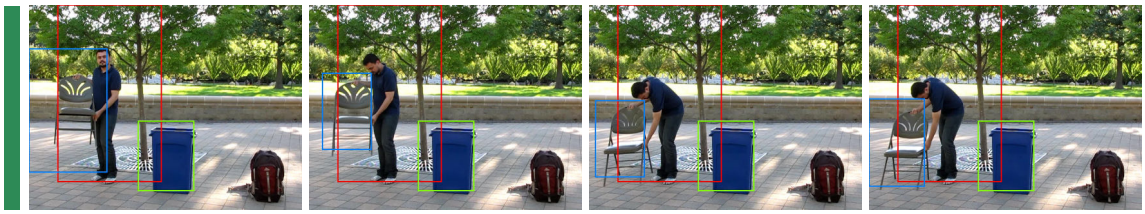
Figure 22: Sentential descriptions generated for each of the 94 video clips in the first corpus subject to the contraction threshold 0.90. The highest-scoring sentence for each clip is generated, among all sentences that are generated by the grammar in Table 11(a), by means of a beam search. The sentences deemed by human judges to describe the associated clips are indicated in green, while ones that do not are indicated in red.

The chair to the left of the trash can approached the trash can.

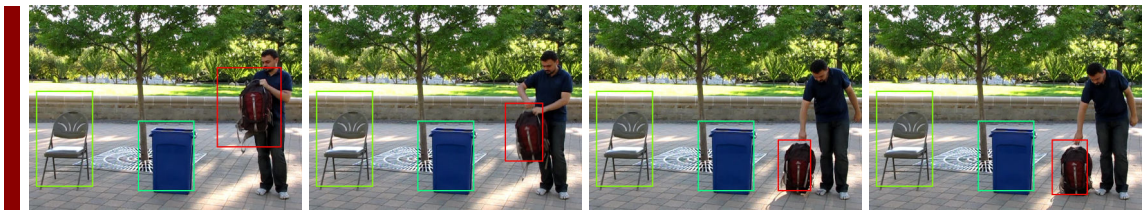


The backpack to the right of the trash can approached the trash can.

The backpack to the right of the trash can approached the trash can.



The person to the left of the trash can put down the chair.

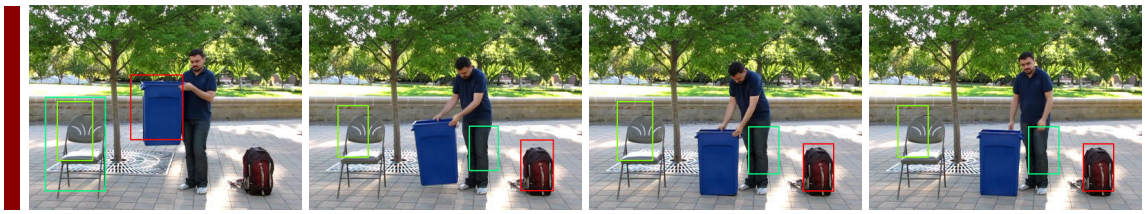


The backpack to the right of the person approached the trash can.

The person to the right of the chair put down the backpack.

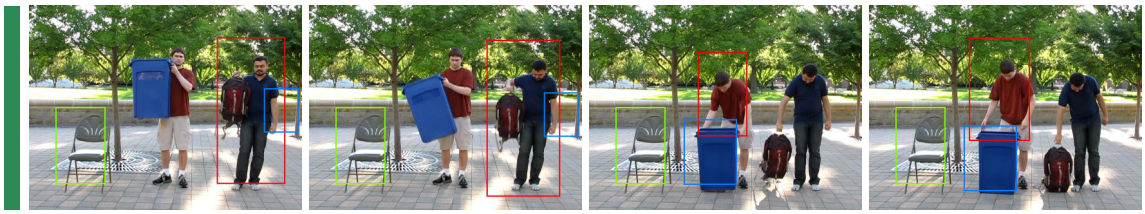
Figure 22: Sentential-description examples continued.

The chair to the left of the trash can approached the backpack.

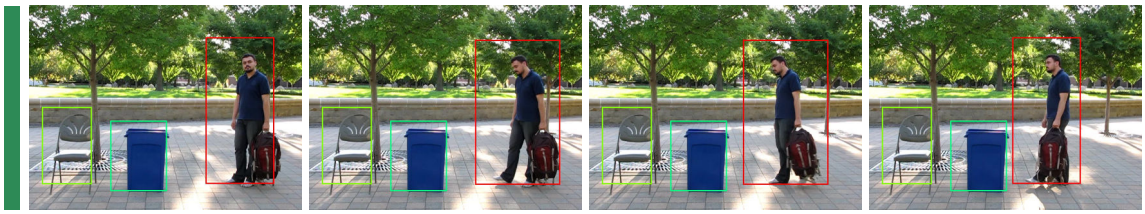


The trash can to the right of the person approached the chair.

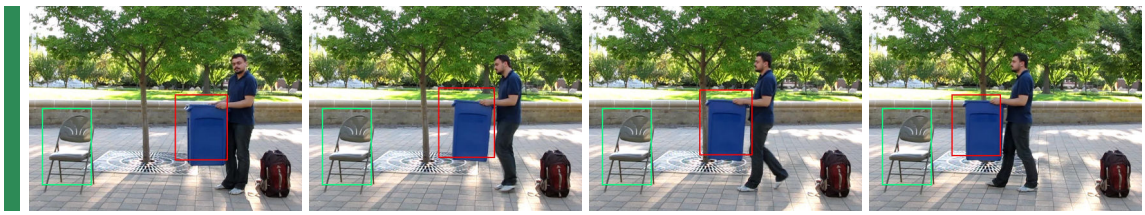
The person to the right of the chair put down the trash can.



The person to the right of the chair put down the trash can.

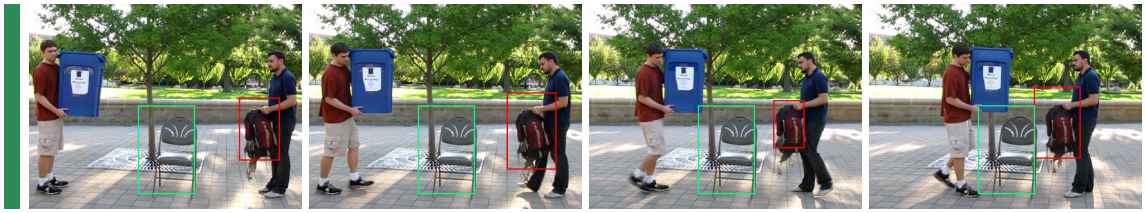


The person to the right of the chair approached the trash can.

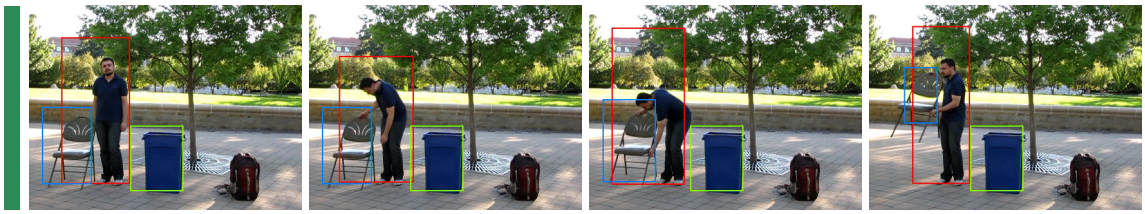


The trash can to the right of the chair approached the chair.

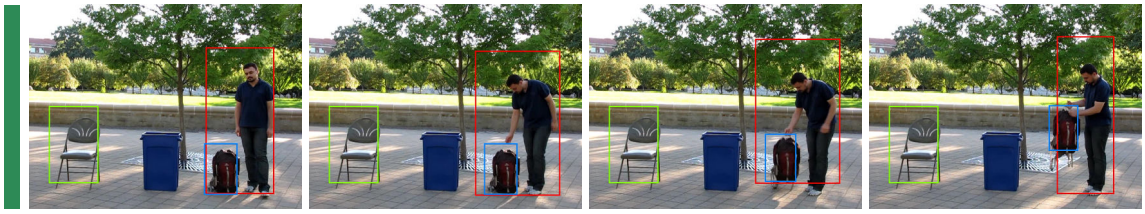
Figure 22: Sentential-description examples continued.



The backpack to the right of the chair approached the chair.



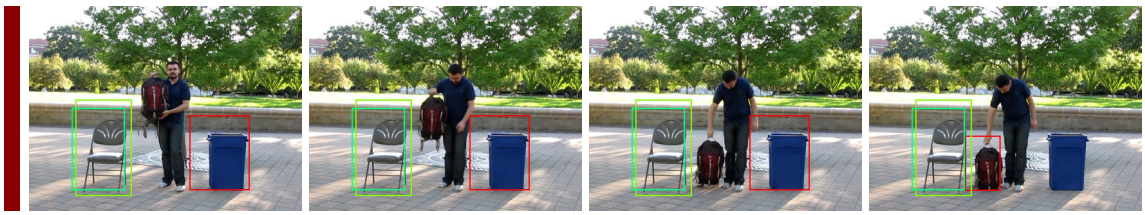
The person to the left of the trash can picked up the chair.



The person to the right of the chair picked up the backpack.

The person to the right of the trash can picked up the backpack.

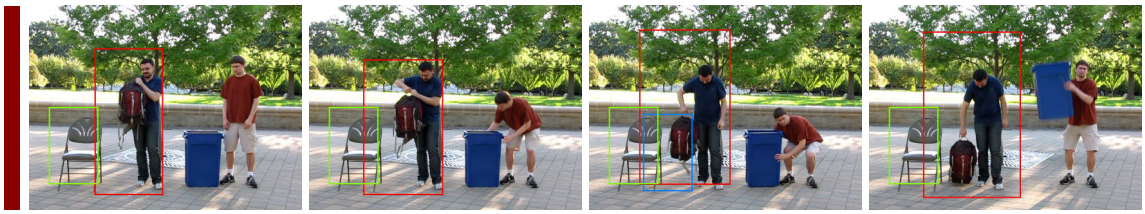
The person to the left of the chair picked up the backpack.



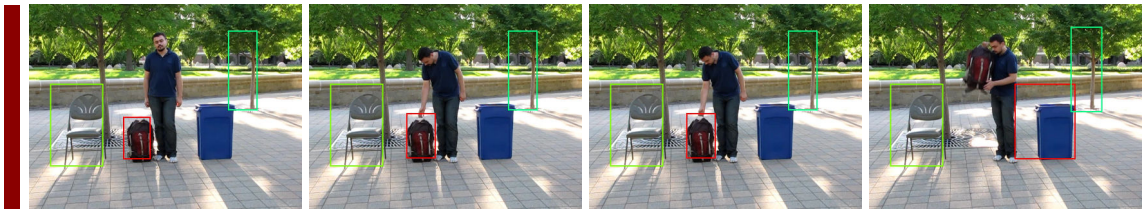
The trash can to the right of the person approached the chair.

Figure 22: Sentential-description examples continued.

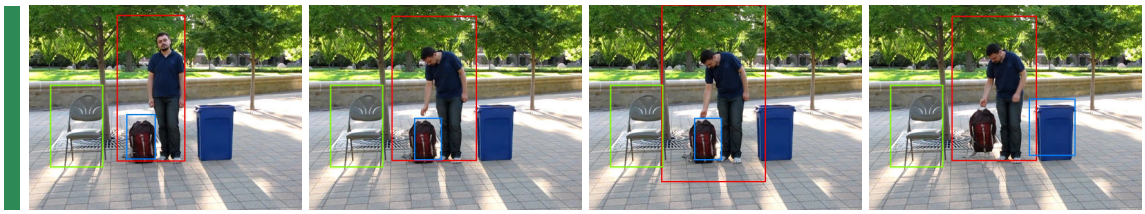
The backpack to the left of the trash can approached the trash can.



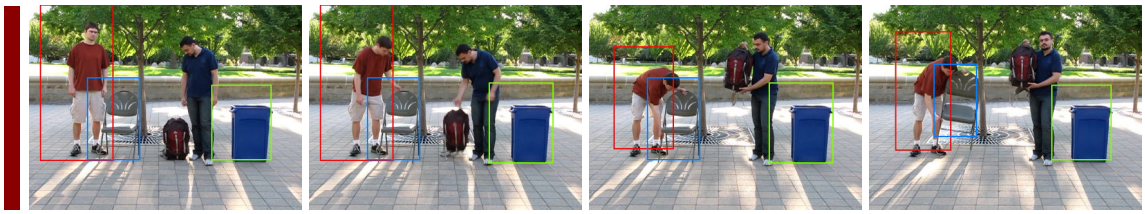
The person to the right of the chair put down the chair.



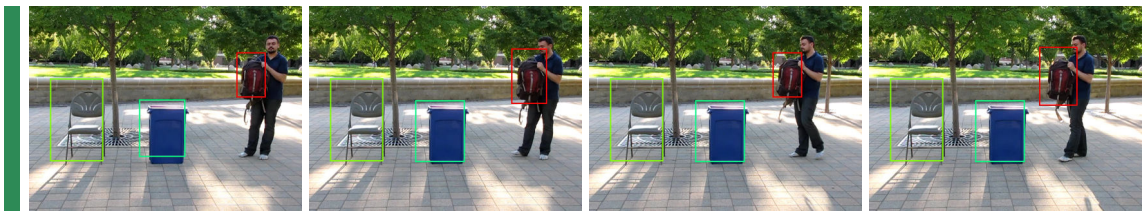
The trash can to the right of the chair approached the person.



The person to the right of the chair picked up the trash can.

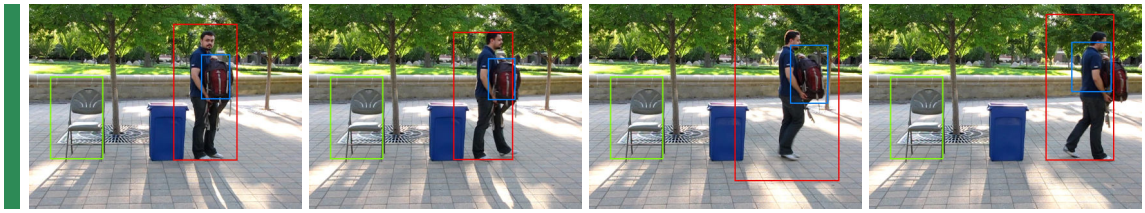


The person to the left of the trash can picked up the chair.

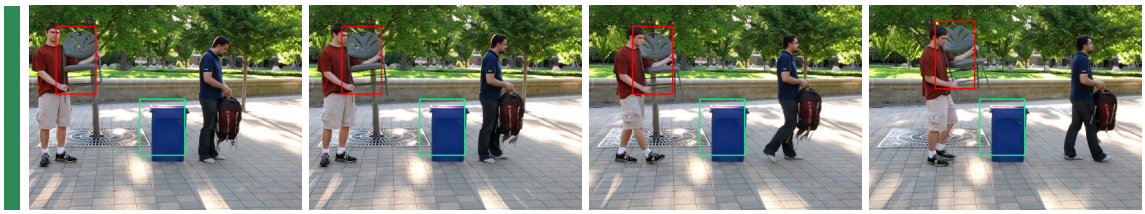


The backpack to the right of the chair approached the trash can.

Figure 22: Sentential-description examples continued.

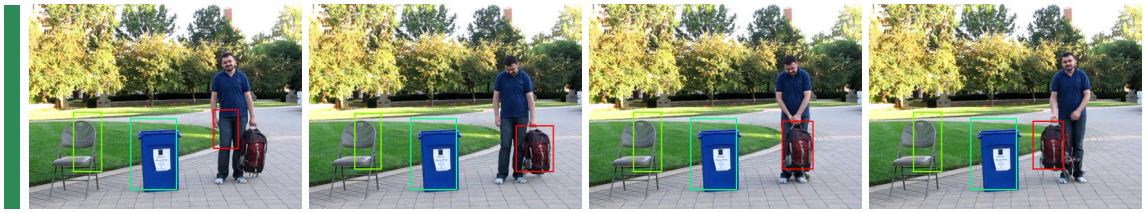


The person to the right of the chair carried the backpack.

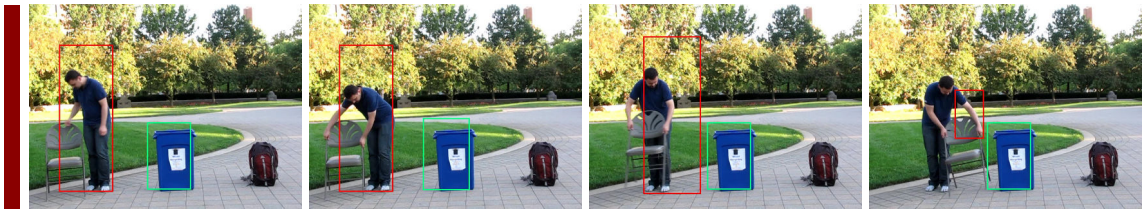


The chair to the left of the trash can approached the trash can.

The person to the right of the chair approached the chair.



The backpack to the right of the person approached the trash can.



The person to the left of the trash can approached the trash can.

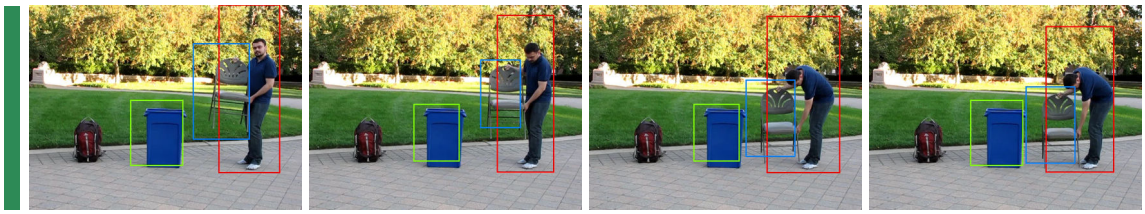
The backpack to the right of the trash can approached the trash can.

Figure 22: Sentential-description examples continued.

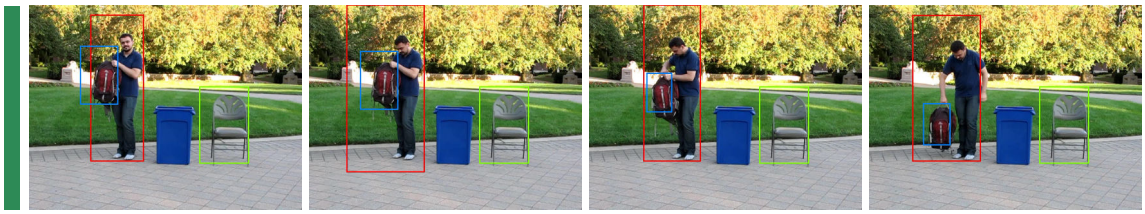
The backpack to the left of the chair approached the chair.

The trash can to the right of the backpack approached the chair.

The trash can to the right of the chair approached the chair.



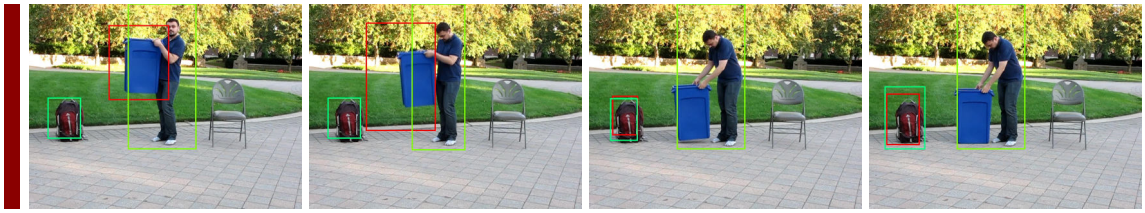
The person to the right of the trash can put down the chair.



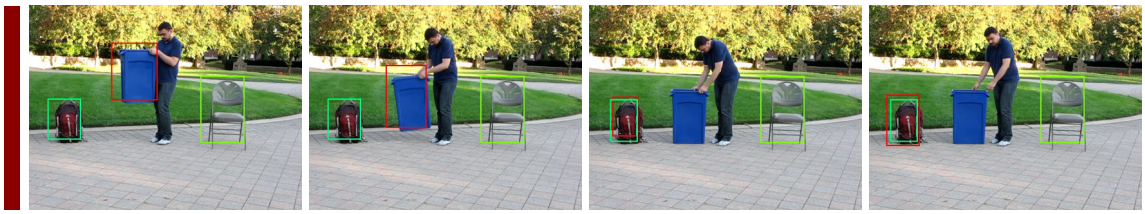
The person to the left of the chair put down the backpack.

The chair to the right of the trash can approached the trash can.

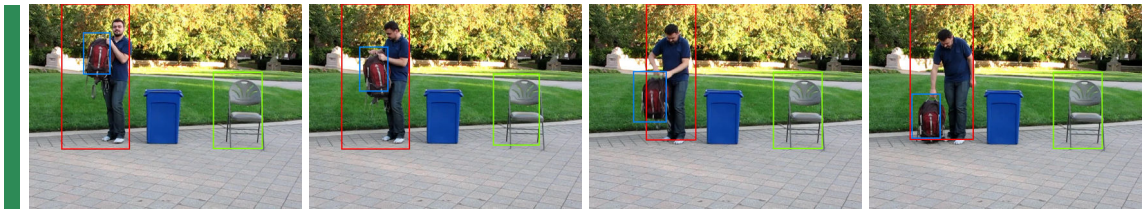
Figure 22: Sentential-description examples continued.



The trash can to the left of the person approached the backpack.

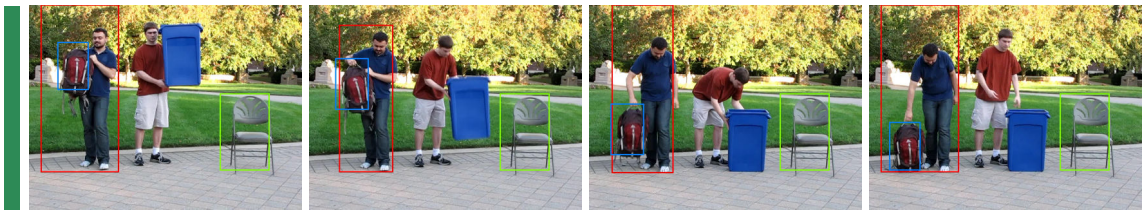


The trash can to the left of the chair approached the backpack.

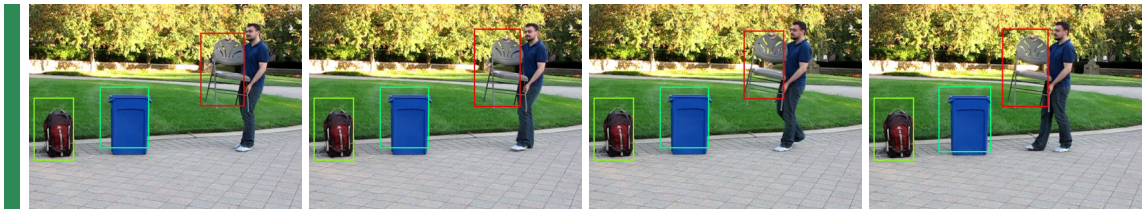


The person to the left of the chair put down the backpack.

The person to the left of the chair put down the backpack.

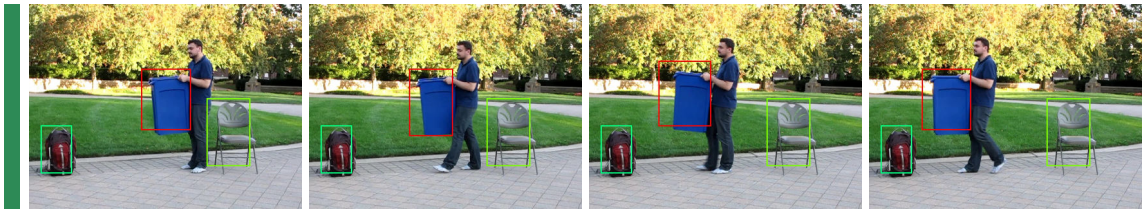


The person to the left of the chair put down the backpack.

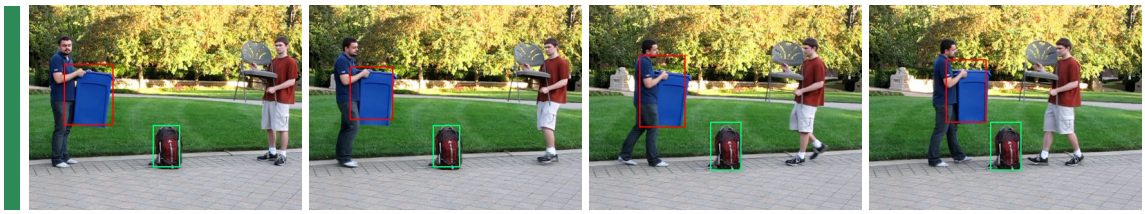


The chair to the right of the backpack approached the trash can.

Figure 22: Sentential-description examples continued.

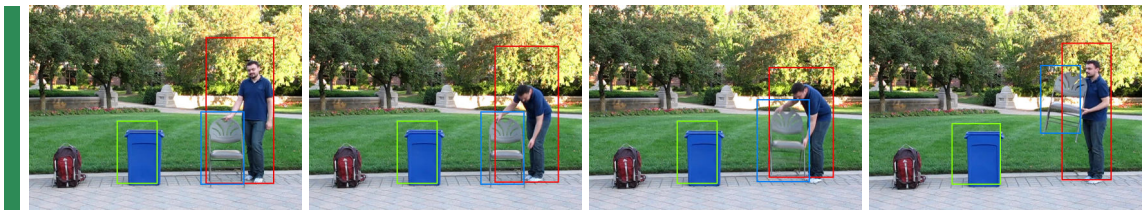


The trash can to the left of the chair approached the backpack.

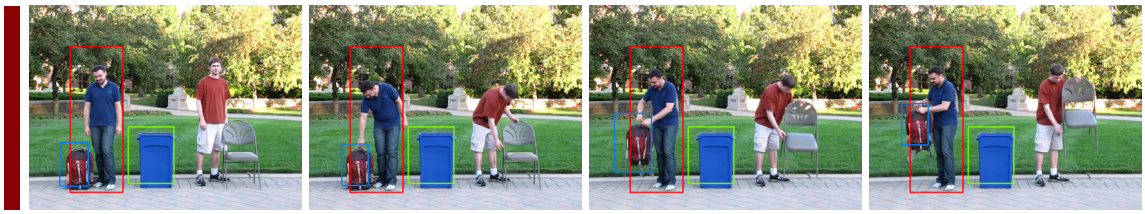


The trash can to the left of the backpack approached the backpack.

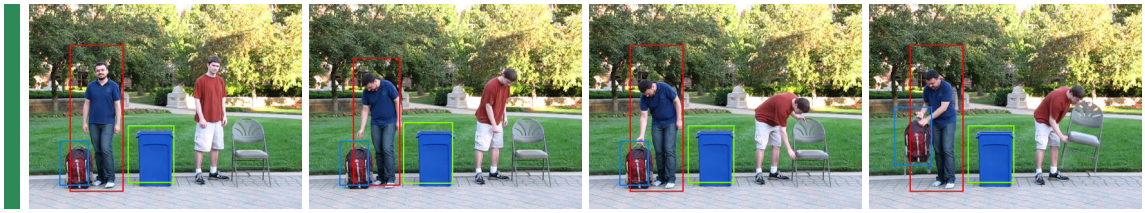
The backpack to the left of the chair approached the trash can.



The person to the right of the trash can picked up the chair.



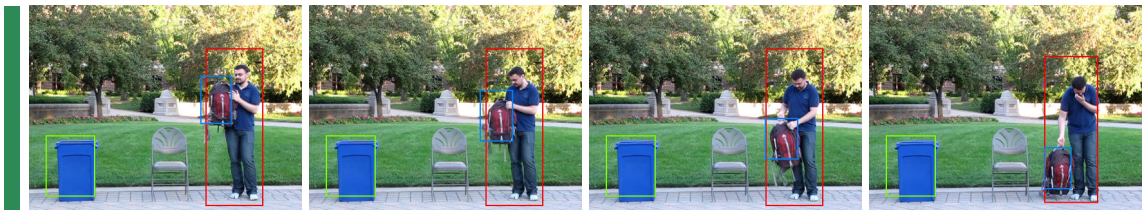
The person to the left of the trash can picked up the trash can.



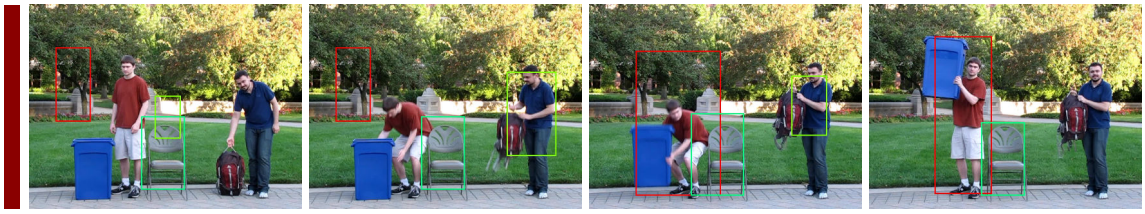
The person to the left of the trash can picked up the backpack.

Figure 22: Sentential-description examples continued.

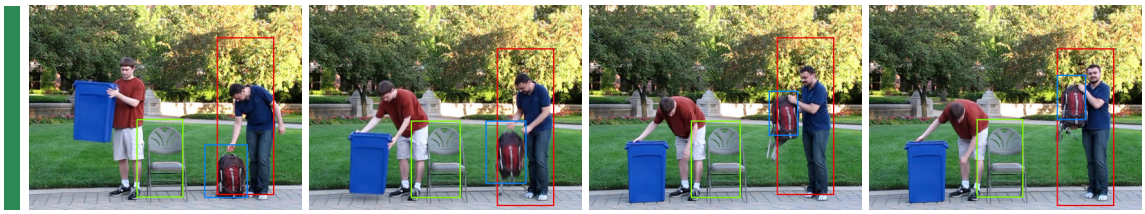
The person to the right of the chair picked up the backpack.



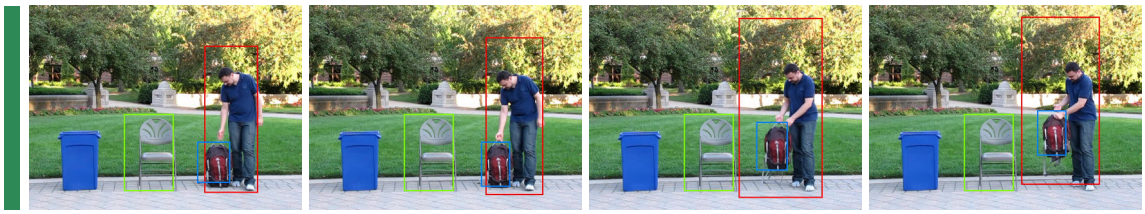
The person to the right of the trash can put down the backpack.



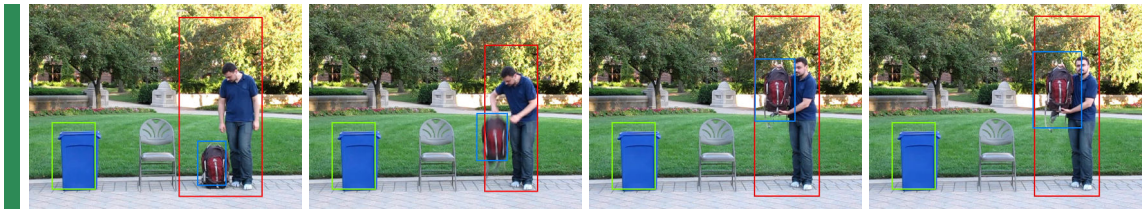
The person to the left of the chair approached the chair.



The person to the right of the chair picked up the backpack.

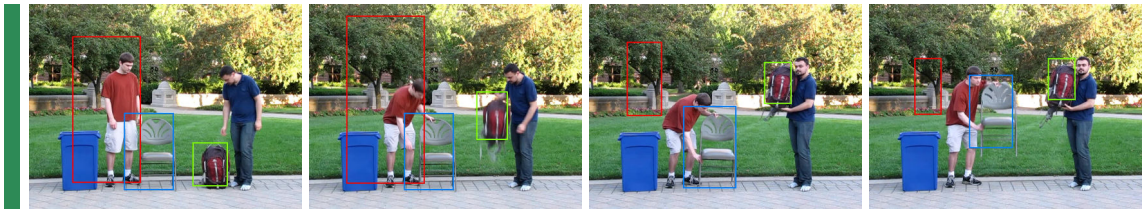


The person to the right of the chair picked up the backpack.

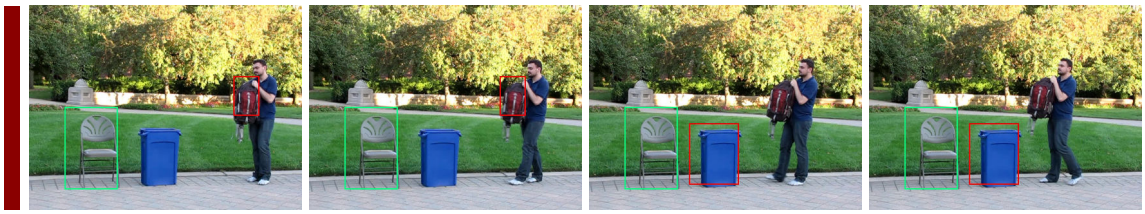


The person to the right of the trash can picked up the backpack.

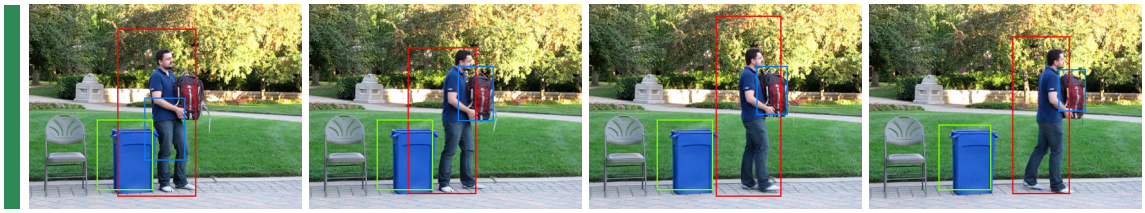
Figure 22: Sentential-description examples continued.



The person to the left of the backpack picked up the chair.

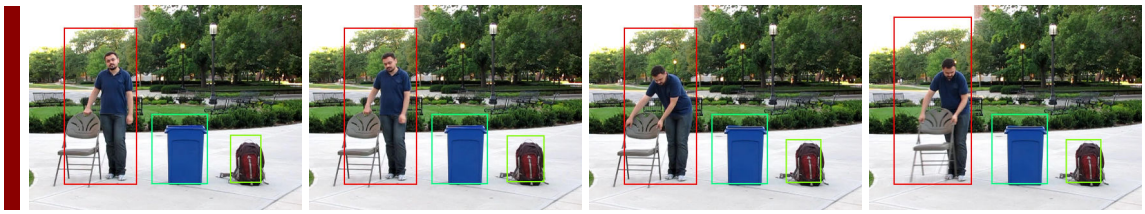


The trash can to the right of the chair approached the chair.

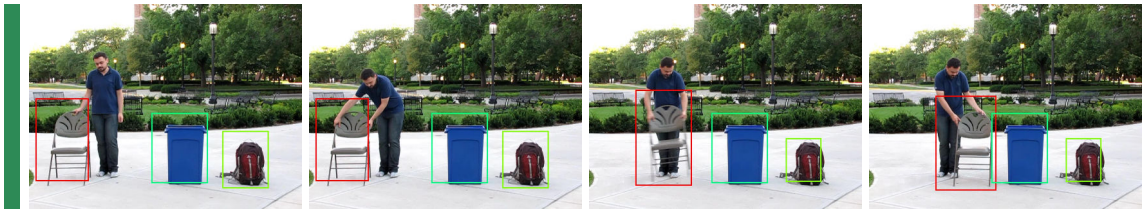


The person to the right of the trash can carried the backpack.

The chair to the left of the trash can approached the trash can.

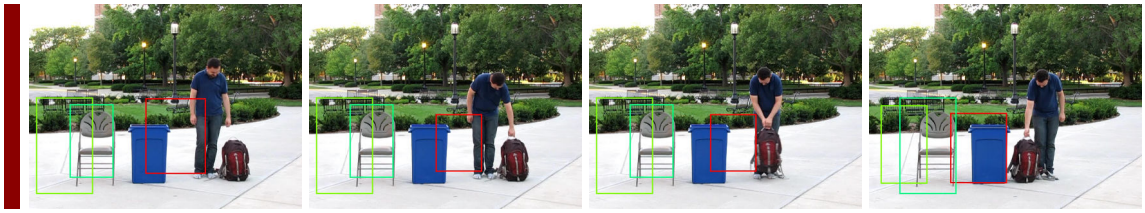


The person to the left of the backpack approached the trash can.



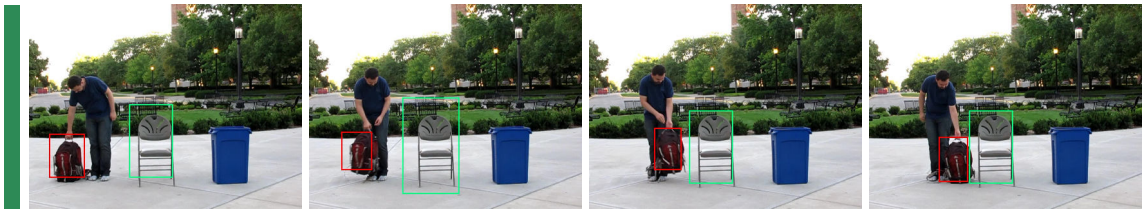
The chair to the left of the backpack approached the trash can.

Figure 22: Sentential-description examples continued.

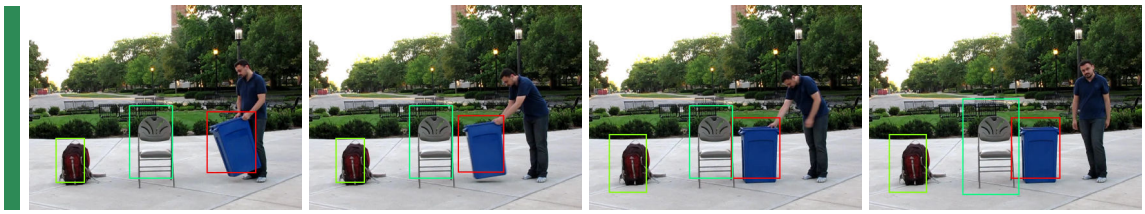


The trash can to the right of the person approached the chair.

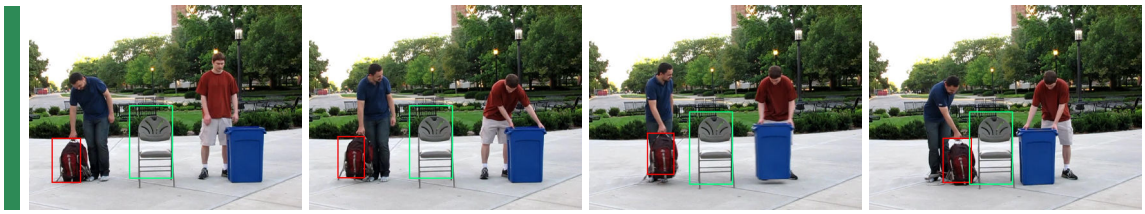
The backpack to the right of the trash can approached the trash can.



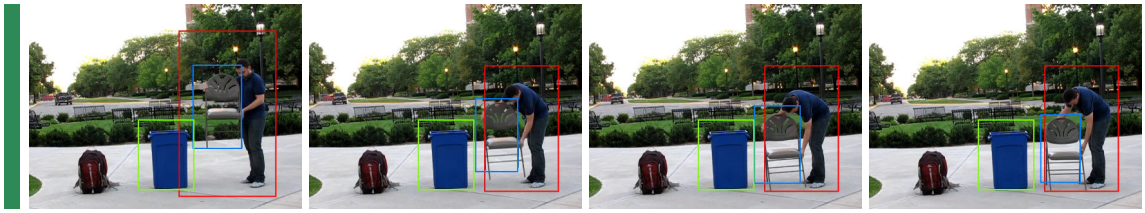
The backpack to the left of the chair approached the chair.



The trash can to the right of the backpack approached the chair.

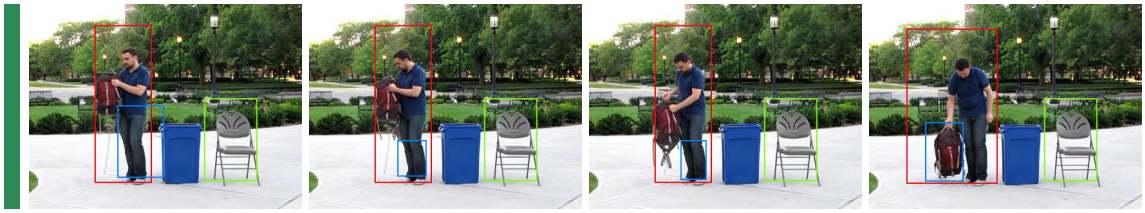


The backpack to the left of the chair approached the chair.



The person to the right of the trash can put down the chair.

Figure 22: Sentential-description examples continued.

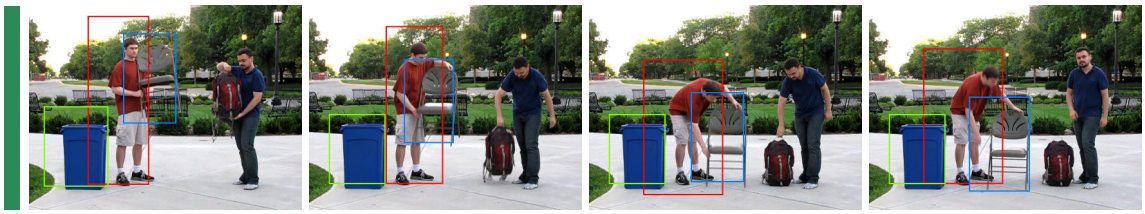


The person to the left of the chair put down the backpack.

The person to the left of the trash can put down the backpack.

The person to the right of the chair put down the backpack.

The person to the right of the chair put down the chair.

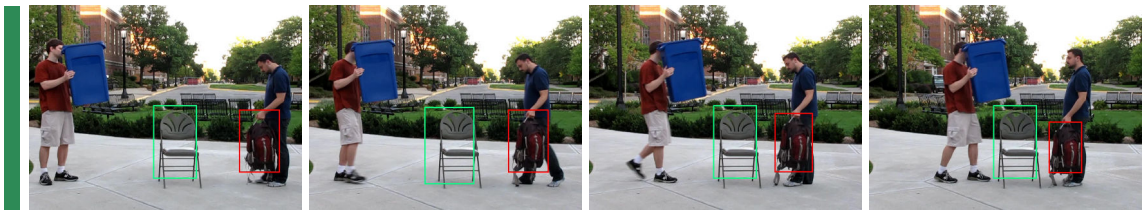


The person to the right of the trash can put down the chair.

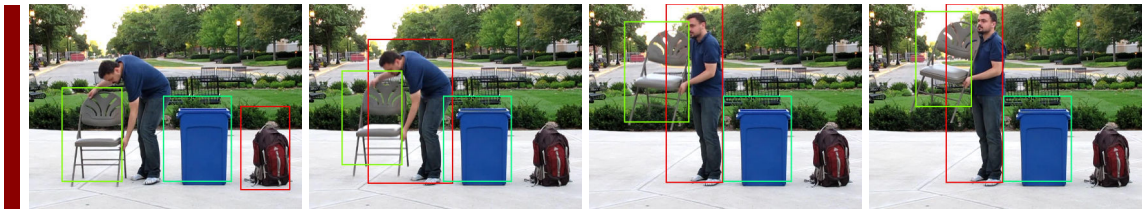
The backpack to the right of the trash can approached the chair.

Figure 22: Sentential-description examples continued.

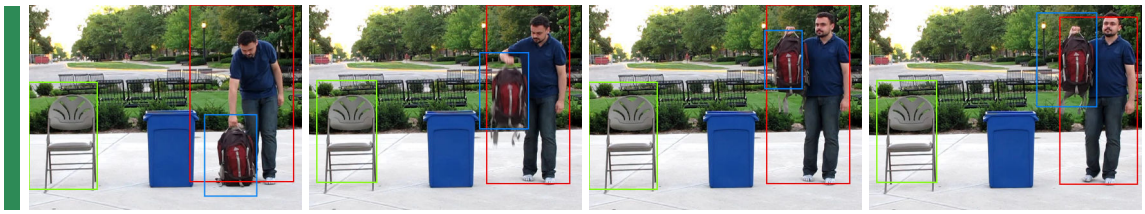
The person to the left of the backpack carried the trash can.



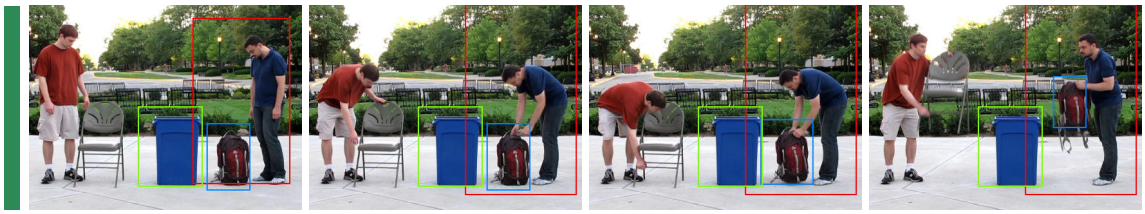
The backpack to the right of the chair approached the chair.



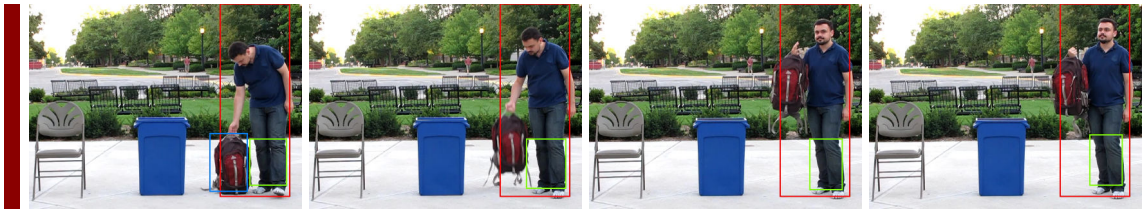
The person to the right of the chair approached the trash can.



The person to the right of the chair picked up the backpack.



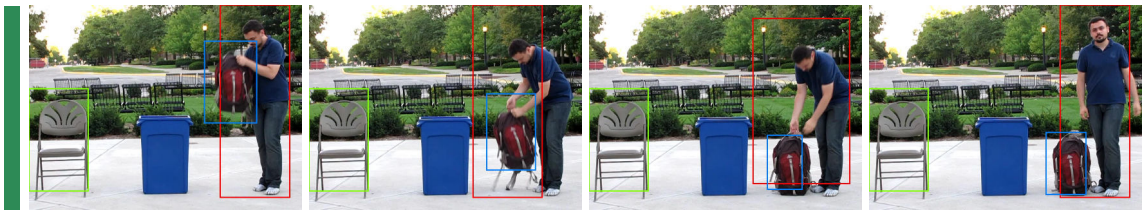
The person to the right of the trash can picked up the backpack.



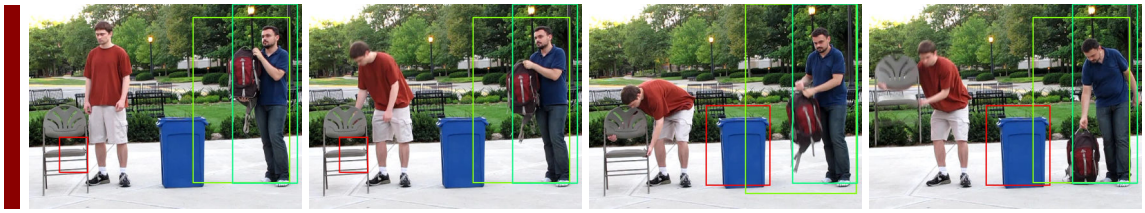
The person to the left of the backpack picked up the backpack.

Figure 22: Sentential-description examples continued.

The trash can to the right of the chair approached the person.



The person to the right of the chair put down the backpack.



The trash can to the left of the person approached the person.

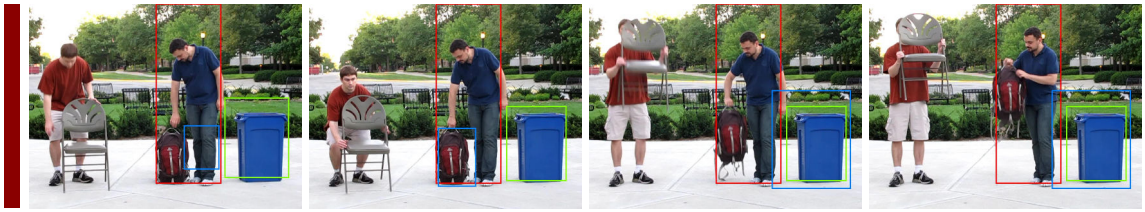
The person to the right of the chair picked up the backpack.

The person to the left of the chair put down the trash can.

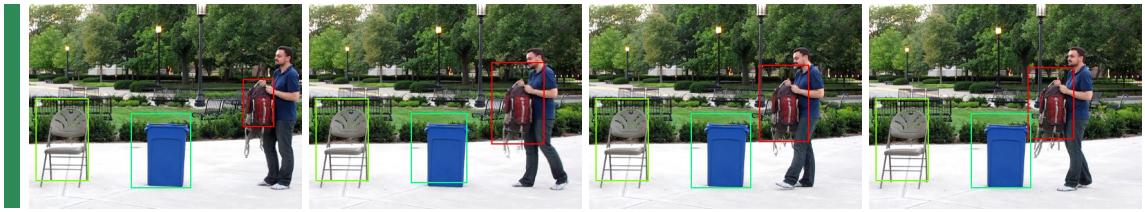
The person to the left of the trash can picked up the trash can.

Figure 22: Sentential-description examples continued.

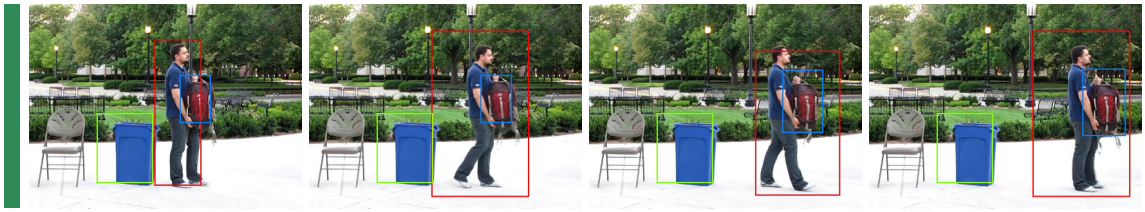
The person to the left of the trash can picked up the backpack.



The person to the left of the trash can picked up the trash can.



The backpack to the right of the chair approached the trash can.



The person to the right of the trash can carried the backpack.

The chair to the left of the trash can approached the trash can.

Figure 22: Sentential-description examples continued.

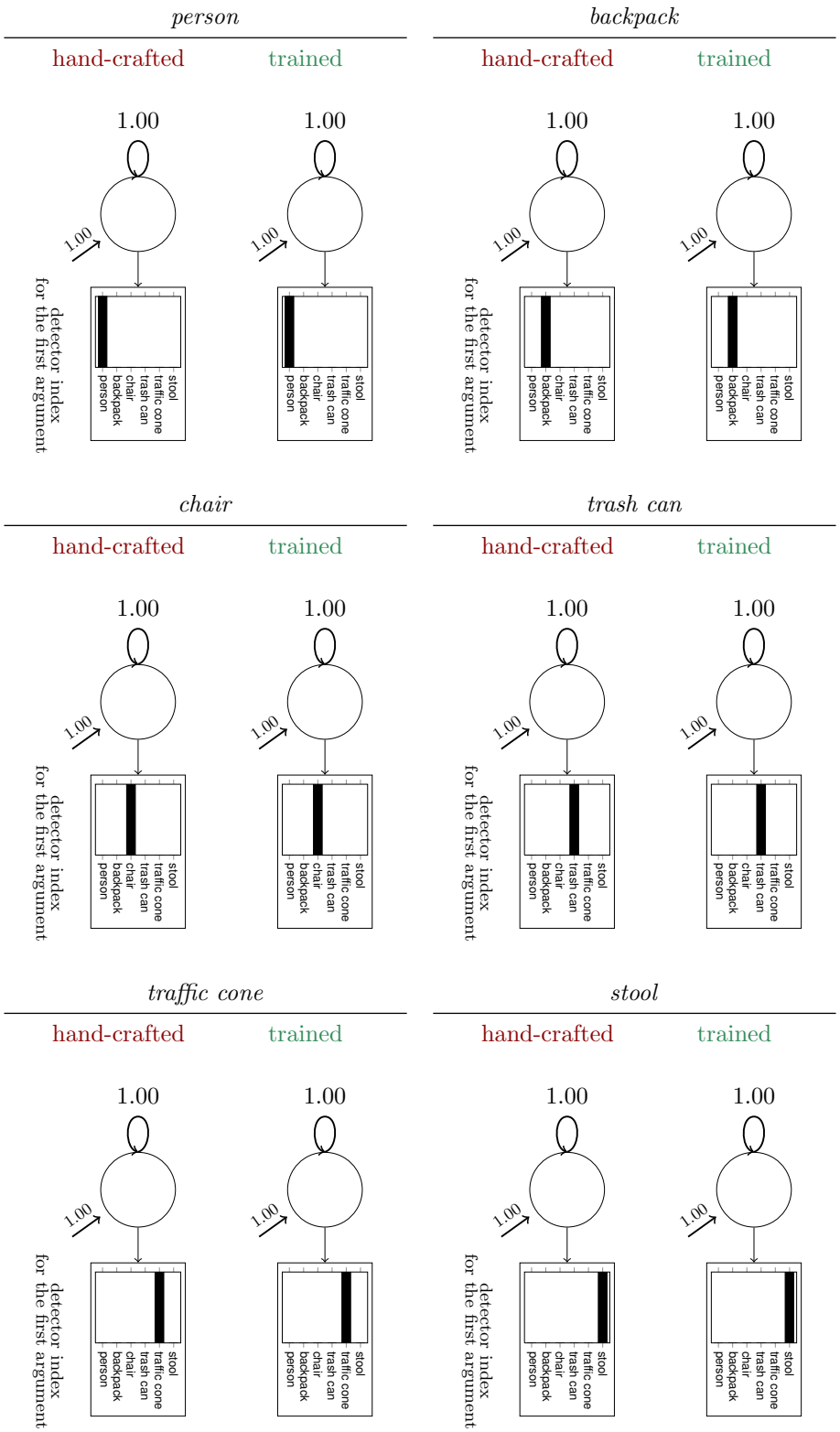


Figure 23: Comparison between hand-crafted and trained models for nouns.

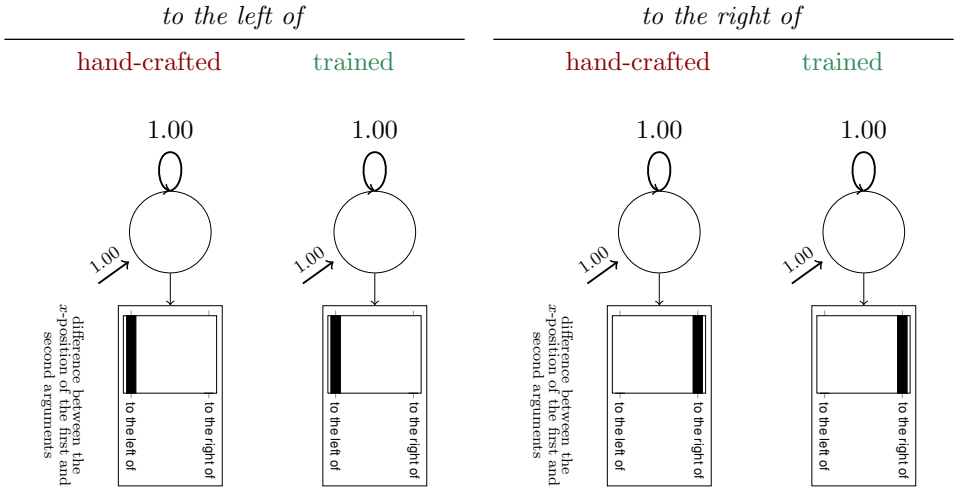


Figure 24: Comparison between hand-crafted and trained models for spatial-relation prepositions.

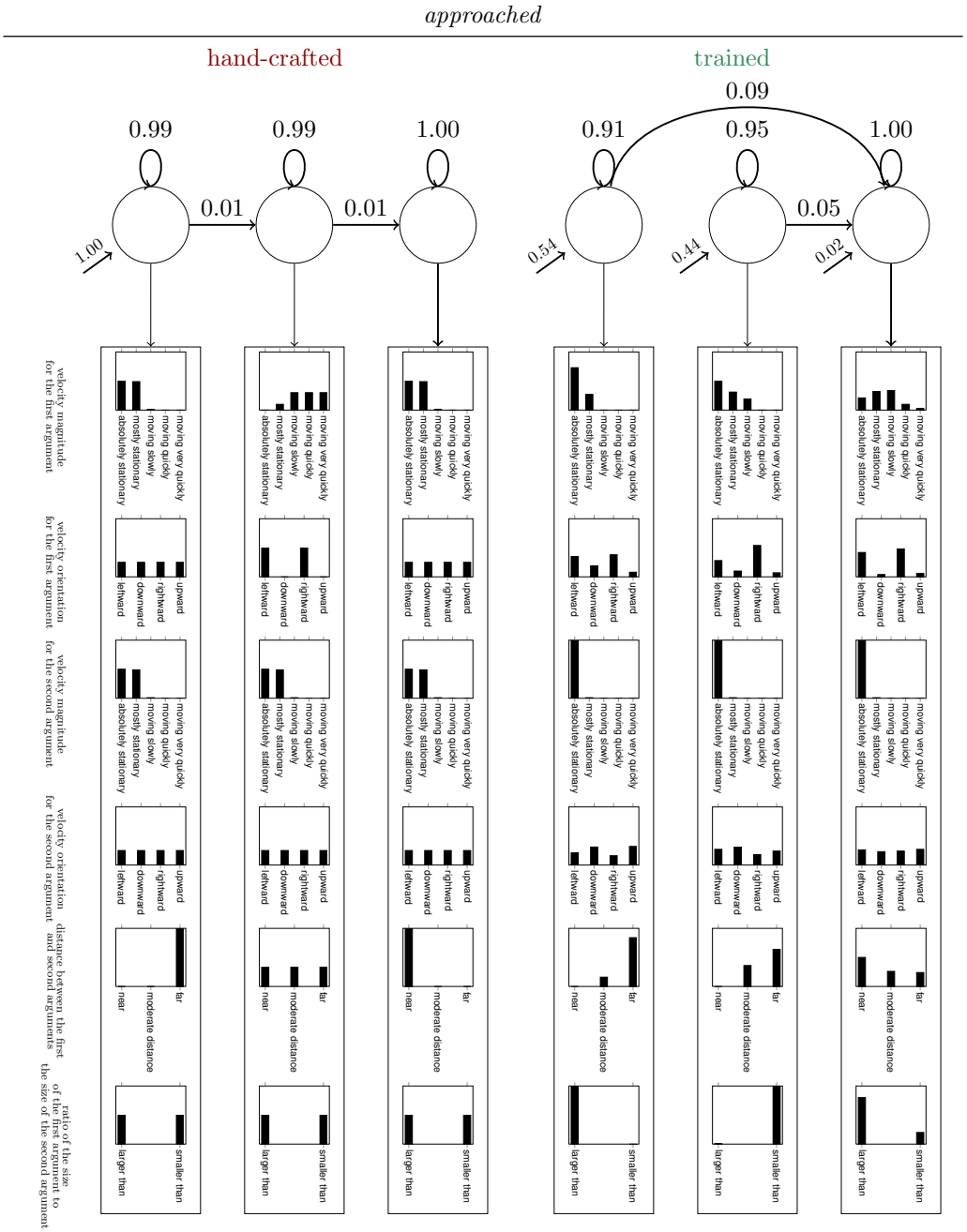


Figure 25: Comparison between hand-crafted and trained models for the verb *approached*.

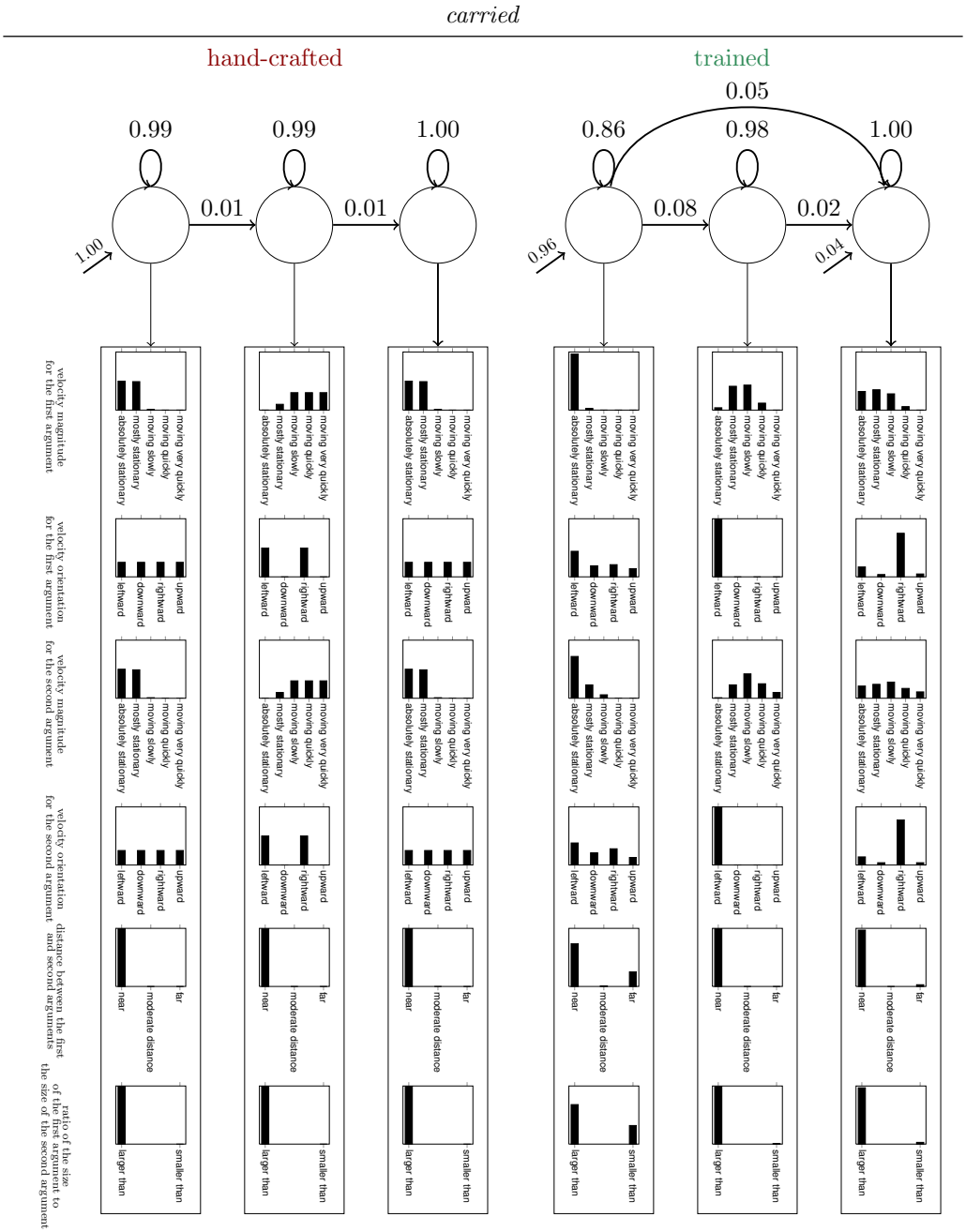


Figure 26: Comparison between hand-crafted and trained models for the verb *carried*.

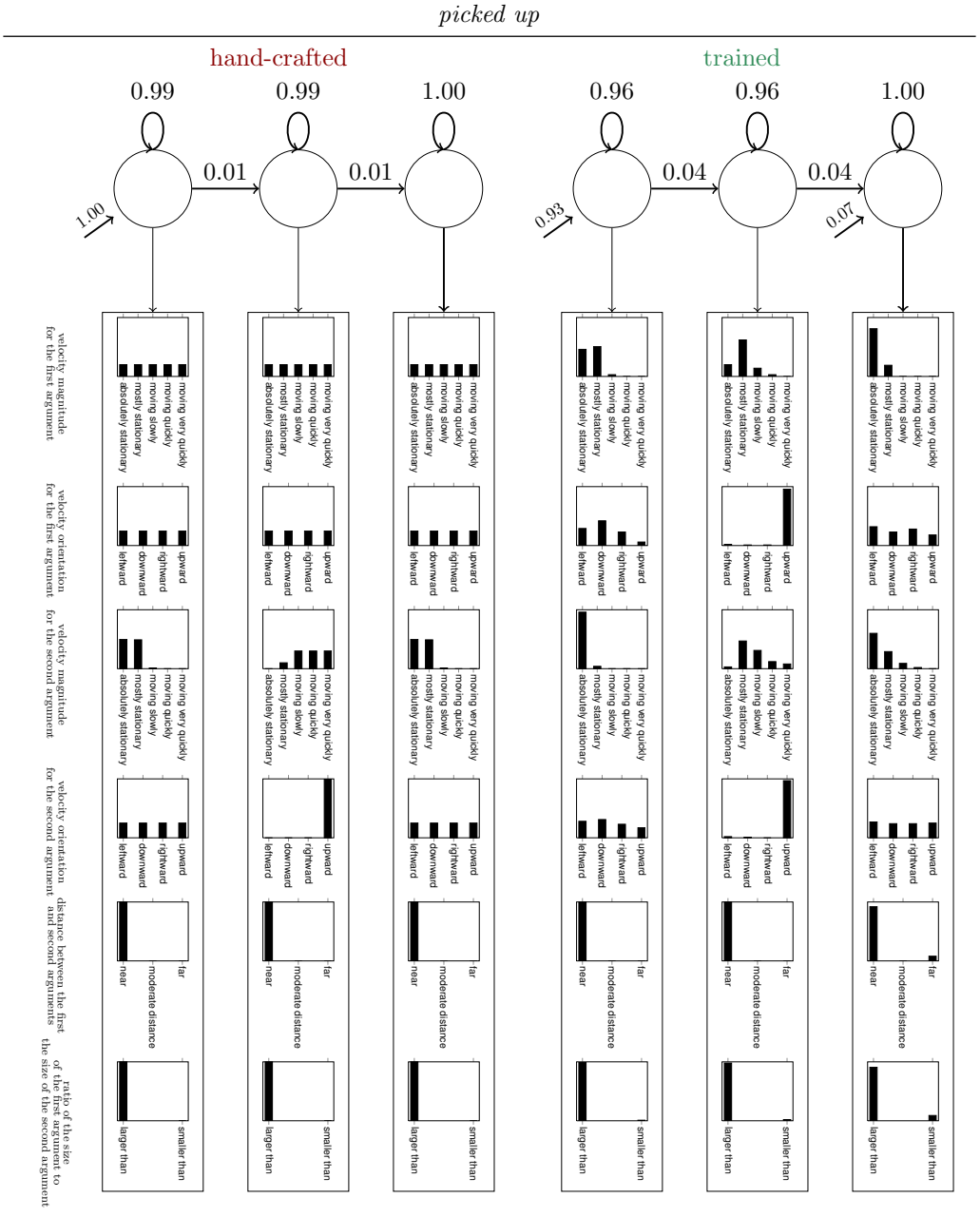


Figure 27: Comparison between hand-crafted and trained models for the verb *picked up*.

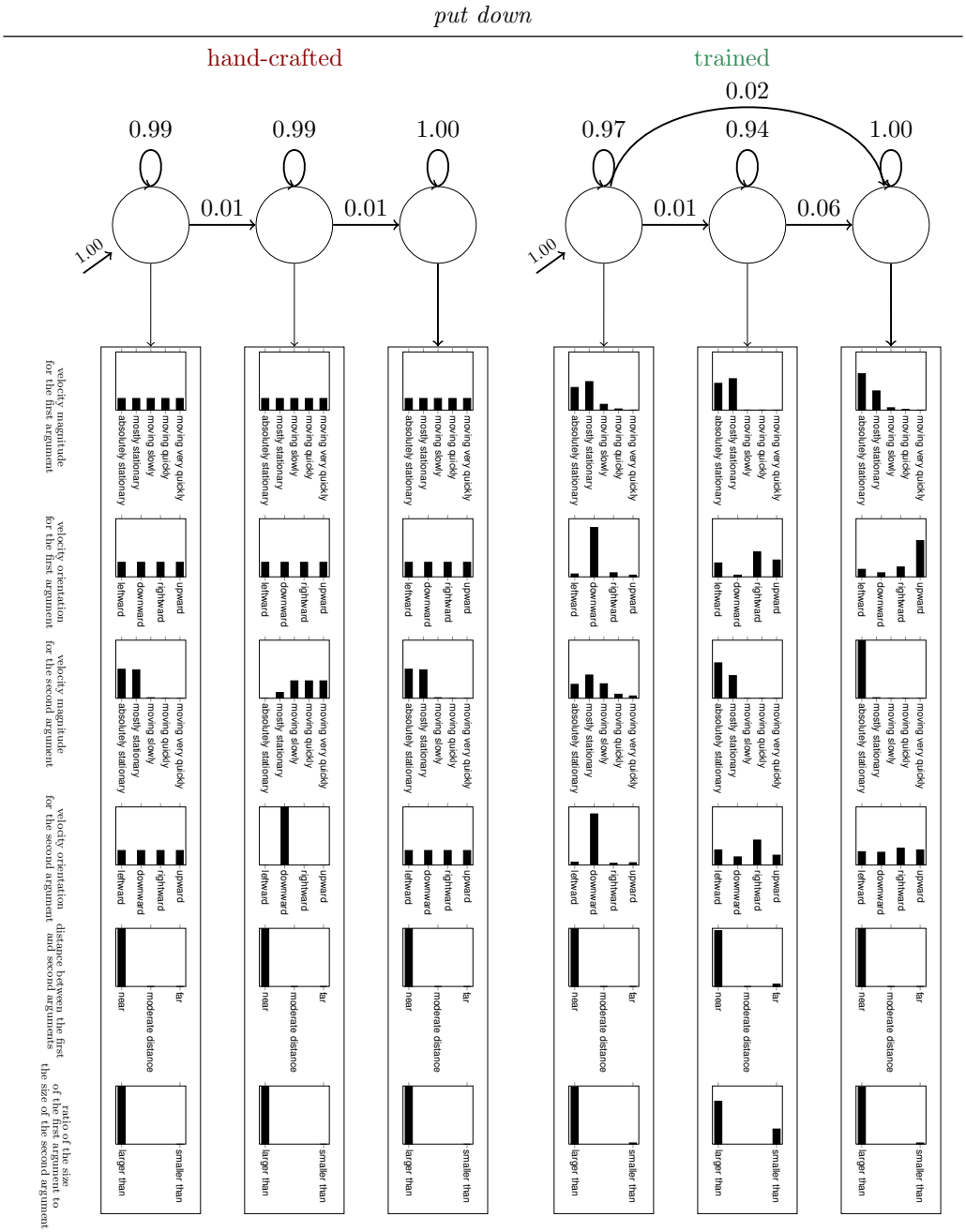


Figure 28: Comparison between hand-crafted and trained models for the verb *put down*.

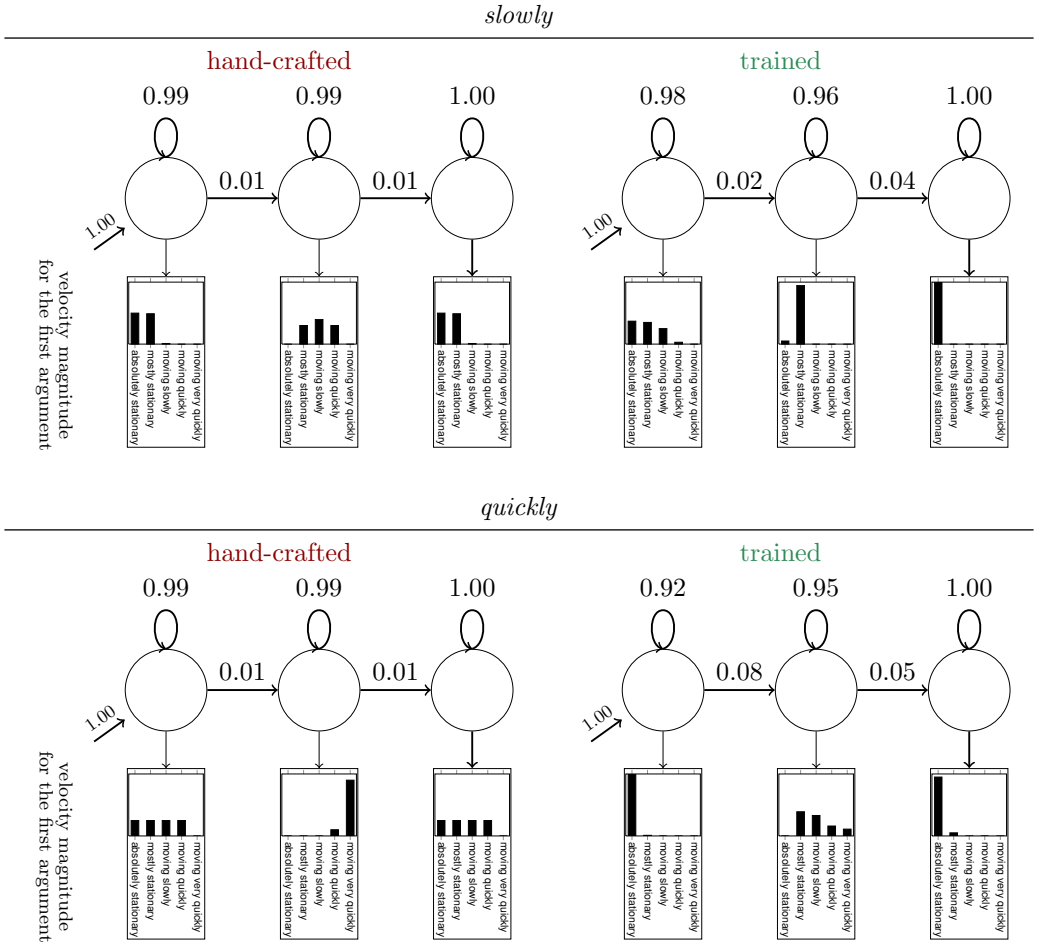


Figure 29: Comparison between hand-crafted and trained models for adverbs.

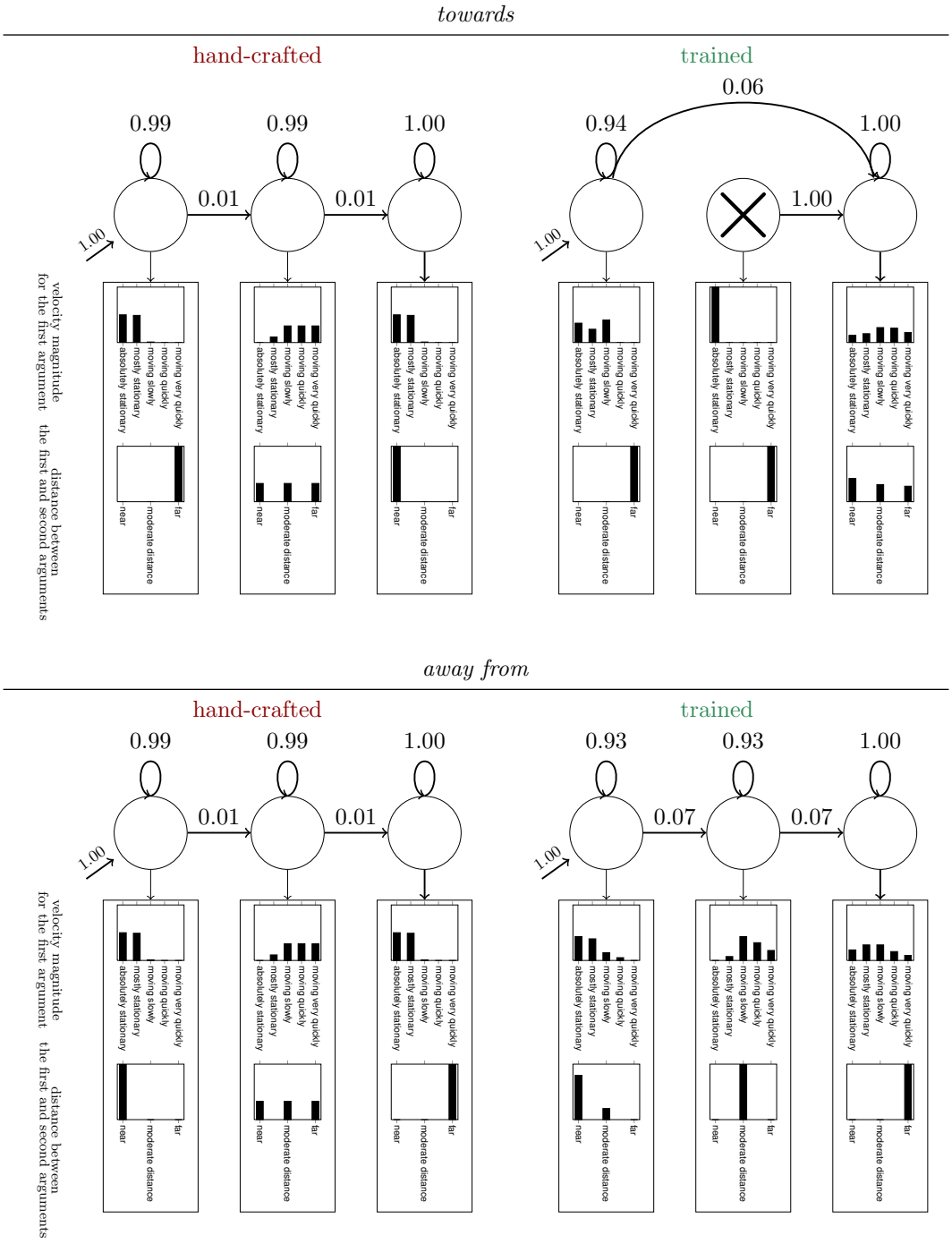


Figure 30: Comparison between hand-crafted and trained models for motion prepositions.

References

Aoun, J., & Sportiche, D. (1983). On the formal theory of government. *Linguistic Review*, *2*(3), 211–236.

Avidan, S. (2004). Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(8), 1064–1072.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., & Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, *92*(1), 1–31.

Barbu, A., Barrett, D. P., Chen, W., Siddharth, N., Xiong, C., Corso, J. J., Fellbaum, C. D., Hanson, C., Hanson, S. J., Hélie, S., Malaia, E., Pearlmutter, B. A., Siskind, J. M., Talavage, T. M., & Wilbur, R. B. (2014). Seeing is worse than believing: Reading people's minds better than computer-vision methods recognize actions. In *Proceedings of the European Conference on Computer Vision*, pp. 612–627.

Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., Fidler, S., Michaux, A., Mussman, S., Siddharth, N., Salvi, D., Schmidt, L., Shangguan, J., Siskind, J. M., Waggoner, J., Wang, S., Wei, J., Yin, Y., & Zhang, Z. (2012a). Video in sentences out. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 102–112.

Barbu, A., Siddharth, N., Michaux, A., & Siskind, J. M. (2012b). Simultaneous object detection, tracking, and event recognition. *Advances in Cognitive Systems*, *2*, 203–220.

Barbu, A., Siddharth, N., & Siskind, J. M. (2014). Language-driven video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop on Vision Meets Cognition*.

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, *3*, 1–8.

Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, *37*(6), 1554–1563.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, *41*(1), 164–171.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Berclaz, J., Fleuret, F., Turetken, E., & Fua, P. (2011). Multiple object tracking using K-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(9), 1806–1819.

Bilmes, J. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Tech. rep. TR-97-021, ICSI.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., & Basri, R. (2005). Actions as space-time shapes. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1395–1402.

Börschinger, B., Jones, B. K., & Johnson, M. (2011). Reducing grounded learning tasks to grammatical inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1416–1425.

Brand, M., Oliver, N., & Pentland, A. (1997). Coupled hidden Markov models for complex action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 994–999.

Chen, C.-Y., & Grauman, K. (2013). Watching unlabeled videos helps learn new human actions from very few labeled snapshots. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 572–579.

Chen, D. L., & Mooney, R. J. (2008). Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the International Conference on Machine Learning*, pp. 128–135.

Chen, D. L., & Mooney, R. J. (2011). Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Conference on Artificial Intelligence*, pp. 859–865.

Chomsky, N. (1982). *Some Concepts and Consequences of the Theory of Government and Binding*. MIT Press.

Chomsky, N. (2002). *Syntactic Structures* (Second edition). Walter de Gruyter.

Das, P., Xu, C., Doell, R. F., & Corso, J. J. (2013). A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2634–2641.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, *39*(1), 1–38.

Dinkelbach, W. (1967). On nonlinear fractional programming. *Management Science*, *13*(7), 492–498.

Dominey, P. F., & Boucher, J.-D. (2005). Learning to talk about events from narrated video in a construction grammar framework. *Artificial Intelligence*, *167*(1-2), 31–61.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Everts, I., van Gemert, J. C., & Gevers, T. (2013). Evaluation of color stips for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2850–2857.

Farhadi, A., Hejrati, M., Sadeghi, M., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010). Every picture tells a story: Generating sentences from images. In *Proceedings of the European Conference on Computer Vision*, pp. 15–29.

Fellbaum, C. (1998). *WordNet: an electronic lexical database*. MIT Press.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010a). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Felzenszwalb, P. F., Girshick, R. B., & McAllester, D. A. (2010b). Cascade object detection with deformable part models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2241–2248.

Feng, S. L., Manmatha, R., & Lavrenko, V. (2004). Multiple Bernoulli relevance models for image and video annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1002–1009.

Fern, A. P., Givan, R. L., & Siskind, J. M. (2002a). Specific-to-general learning for temporal events. In *Proceedings of the Conference on Artificial Intelligence*, pp. 152–158.

Fern, A. P., Givan, R. L., & Siskind, J. M. (2002b). Specific-to-general learning for temporal events with application to learning event definitions from video. *Journal of Artificial Intelligence Research*, *17*, 379–449.

Fern, A. P., Siskind, J. M., & Givan, R. L. (2002c). Learning temporal, relational, force-dynamic event definitions from video. In *Proceedings of the Conference on Artificial Intelligence*, pp. 159–166.

Fernández Tena, C., Baiget, P., Roca, X., & Gonzàlez, J. (2007). Natural language descriptions of human behavior from video sequences. In *Advances in Artificial Intelligence*, pp. 279–292.

Gaidon, A., Harchaoui, Z., & Schmid, C. (2014). Activity representation with motion hierarchies. *International Journal of Computer Vision*, *107*(3), 219–238.

Grimshaw, J. (1979). Complement selection and the lexicon. *Linguistic Inquiry*, *10*(2), 279–326.

Grimshaw, J. (1981). Form, function, and the language acquisition device. In Baker, C. L., & McCarthy, J. J. (Eds.), *The Logical Problem of Language Acquisition*, pp. 165–182. MIT Press.

Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., & Saenko, K. (2013). Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2712–2719.

Gupta, A., Verma, Y., & Jawahar, C. (2012). Choosing linguistics over vision to describe images. In *Proceedings of the Conference on Artificial Intelligence*, pp. 606–612.

Haegeman, L. (1992). *Introduction to government and binding theory*. Blackwell.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, *29*(2), 147–160.

Han, M., Sethi, A., Hua, W., & Gong, Y. (2004). A detection-based multiple object tracking method. In *Proceedings of the IEEE International Conference on Image Processing*, pp. 3065–3068.

Hanckmann, P., Schutte, K., & Burghouts, G. J. (2012). Automated textual descriptions for a wide range of video events with 48 human actions. In *Proceedings of the European Conference on Computer Vision (Workshops and Demonstrations)*, pp. 372–380.

Ikizler-Cinbis, N., & Sclaroff, S. (2010). Object, scene and actions: Combining multiple features for human action recognition. In *Proceedings of the European Conference on Computer Vision*, pp. 494–507.

Jackendoff, R. (1977). *X-bar-syntax: A study of phrase structure*. MIT Press.

Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, *30*(1), 175–193.

Jie, L., Caputo, B., & Ferrari, V. (2009). Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In *Proceedings of the Neural Information Processing Systems Conference*, pp. 1168–1176.

Khan, M. U. G., & Gotoh, Y. (2012). Describing video contents in natural language. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pp. 27–35.

Khan, M. U. G., Zhang, L., & Gotoh, Y. (2011). Human focused video description. In *Proceedings of the IEEE International Conference on Computer Vision (Workshops)*, pp. 1480–1487.

Kim, J., & Mooney, R. J. (2010). Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the International Conference on Computational Linguistics*, pp. 543–551.

Kim, J., & Mooney, R. J. (2012). Unsupervised PCFG induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 433–444.

Kim, J., & Mooney, R. J. (2013). Adapting discriminative reranking to grounded language learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 218–227.

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 423–430.

Kojima, A., Tamura, T., & Fukunaga, K. (2002). Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, *50*(2), 171–184.

Krishnamoorthy, N., Malkarnenkar, G., Mooney, R. J., Saenko, K., & Guadarrama, S. (2013). Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the NAACL HLT Workshop on Vision and Language*, pp. 10–19.

Krishnamurthy, J., & Kollar, T. (2013). Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, *1*, 193–206.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: A large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2556–2563.

Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., & Berg, T. L. (2011). Baby talk: Understanding and generating simple image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1601–1608.

Kuznetsova, P., Ordonez, V., Berg, A. C., Berg, T. L., & Choi, Y. (2012). Collective generation of natural image descriptions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 359–368.

Kwiatkowski, T., Goldwater, S., Zettlemoyer, L., & Steedman, M. (2012). A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 234–244.

Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., & Steedman, M. (2010). Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1223–1233.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2/3), 107–123.

Li, P., & Ma, J. (2011). What is happening in a still picture?. In *Proceedings of the Asian Conference on Pattern Recognition*, pp. 32–36.

Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1996–2003.

Mann, R., Jepson, A. D., & Siskind, J. M. (1996). The computational perception of scene dynamics. In *Proceedings of the European Conference on Computer Vision*, pp. 528–539.

Mann, R., Jepson, A. D., & Siskind, J. M. (1997). The computational perception of scene dynamics. *Computer Vision and Image Understanding*, *65*(2), 113–128.

Marocco, D., Cangelosi, A., Fischer, K., & Belpaeme, T. (2010). Grounding action words in the sensorimotor interaction with the world: experiments with a simulated iCub humanoid robot. *Frontiers in Neurorobotics*, *4*(7), 1–15.

Messing, R., Pal, C., & Kautz, H. (2009). Activity recognition using the velocity histories of tracked keypoints. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 104–111.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Mitchell, M., Dodge, J., Goyal, A., Yamaguchi, K., Stratos, K., Han, X., Mensch, A., Berg, A. C., Berg, T. L., & III, H. D. (2012). Midge: Generating image descriptions from computer vision detections. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 747–756.

Niebles, J. C., Chen, C.-W., & Fei-Fei, L. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the European Conference on Computer Vision*, pp. 392–405.

Ordonez, V., Kulkarni, G., & Berg, T. L. (2011). Im2text: Describing images using 1 million captioned photographs. In *Proceedings of the Neural Information Processing Systems Conference*, pp. 1143–1151.

Piantadosi, S. T., Goodman, N. D., Ellis, B. A., & Tenenbaum, J. B. (2008). A Bayesian model of the acquisition of compositional semantics. In *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 1620–1625.

Pinker, S. (1984). *Language Learnability and Language Development*. Harvard University Press.

Pirsiavash, H., Ramanan, D., & Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1201–1208.

Reddy, K. K., & Shah, M. (2013). Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, *24*(5), 971–981.

Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

Rohrbach, M., Qin, W., Titov, I., Thater, S., Pinkal, M., & Schiele, B. (2013). Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 433–440.

Roy, D. (2002). Learning visually-grounded words and syntax for a scene description task. *Computer Speech and Language*, *16*(3-4), 353–385.

Roy, D. K., & Pentland, A. P. (2002). Learning words from sights and sounds: A computational model. *Cognitive Science*, *26*(1), 113–146.

Sadanand, S., & Corso, J. J. (2012). Action bank: A high-level representation of activity in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1234–1241.

Schuldt, C., Laptev, I., & Caputo, B. (2004a). Recognizing human actions: a local SVM approach. In *Proceedings of the International Conference on Pattern Recognition*, pp. 32–36.

Schuldt, C., Laptev, I., & Caputo, B. (2004b). Recognizing human actions: A local svm approach. In *Proceedings of the International Conference on Pattern Recognition*, pp. 32–36.

Siddharth, N., Barbu, A., & Siskind, J. M. (2014). Seeing what you're told: Sentence-guided activity recognition in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 732–739.

Siskind, J. M. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, *61*(1-2), 39–91.

Siskind, J. M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research*, *15*, 31–90.

Siskind, J. M., & Morris, Q. (1996). A maximum-likelihood approach to visual event classification. In *Proceedings of the European Conference on Computer Vision*, pp. 347–360.

Siskind, J. M. (1999). Visual event perception. In *Proceedings of the NEC Research Symposium*, pp. 91–154.

Siskind, J. M. (2000). Visual event classification via force dynamics. In *Proceedings of the Conference on Artificial Intelligence*, pp. 149–155.

Siskind, J. M. (2003). Reconstructing force-dynamic models from video sequences. *Artificial Intelligence*, *151*(1-2), 91–154.

Siskind, J. M., & Morris, Q. (1996). A maximum-likelihood approach to visual event classification. In *Proceedings of the European Conference on Computer Vision*, pp. 347–360.

Smith, K., Smith, A. D. M., & Blythe, R. A. (2011). Cross-situational learning: An experimental study of word-learning mechanisms. *Cognitive Science*, *35*(3), 480–498.

Smith, K., Smith, A. D. M., Blythe, R. A., & Vogt, P. (2006). Cross-situational learning: A mathematical approach. In *Proceedings of the International Workshop on the Emergence and Evolution of Linguistic Communication*, pp. 31–44.

Song, H. O., Zickler, S., Althoff, T., Girshick, R., Fritz, M., Geyer, C., Felzenszwalb, P., & Darrell, T. (2012). Sparselet models for efficient multiclass object detection. In *Proceedings of the European Conference on Computer Vision*, pp. 802–815.

Song, Y., Morency, L.-P., & Davis, R. (2013). Action recognition by hierarchical sequence summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3562–3569.

Starner, T., Weaver, J., & Pentland, A. (1998). Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1371–1375.

Tellex, S., Thaker, P., Joseph, J., & Roy, N. (2013). Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*, *0*, 1–17.

Thompson, C. A., & Mooney, R. J. (2003). Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, *18*, 1–44.

Tian, Y., Sukthankar, R., & Shah, M. (2013a). Spatiotemporal deformable part models for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2642–2649.

Tian, Y., Sukthankar, R., & Shah, M. (2013b). Spatiotemporal deformable part models for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2642–2649.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymtotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *13*(2), 260–267.

Wang, H., Kläser, A., Schmid, C., & Liu, C.-L. (2011). Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3169–3176.

Wang, H., Kläser, A., Schmid, C., & Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, *103*(1), 60–79.

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558.

Wang, Z., Guan, G., Qiu, Y., Zhuo, L., & Feng, D. (2013). Semantic context based refinement for news video annotation. *Multimedia Tools and Applications*, *67*(3), 607–627.

Werlberger, M., Pock, T., & Bischof, H. (2010). Motion estimation with non-local total variation regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2464–2471.

Wolf, J. K., Viterbi, A. M., & Dixon, G. S. (1989). Finding the best set of K paths through a trellis with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, *25*(2), 287–296.

Wu, B., & Nevatia, R. (2007). Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, *75*(2), 247–266.

Yamoto, J., Ohya, J., & Ishii, K. (1992). Recognizing human action in time-sequential images using hidden Markov model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379–385.

Yang, Y., Teo, C. L., Daumé III, H., & Aloimonos, Y. (2011). Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 444–454.

Yao, B. Z., Yang, X., Lin, L., Lee, M. W., & Zhu, S.-C. (2010). I2T: Image parsing to text description. *Proceedings of the IEEE*, *98*(8), 1485–1508.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys*, *38*(4), 1–45.

Yu, C., & Ballard, D. H. (2004). On the integration of grounding language and learning objects. In *Proceedings of the Conference on Artificial Intelligence*, pp. 488–493.

Yu, H., & Siskind, J. M. (2013). Grounded language learning from video described with sentences. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 53–63.

Yu, S.-Z. (2010). Hidden semi-Markov models. *Artificial Intelligence*, *174*(2), 215–243.

Yuan, C., Li, X., Hu, W., Ling, H., & Maybank, S. (2013). 3D R transform on spatio-temporal interest points for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 724–730.

Zettlemoyer, L. S., & Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 658–666.

Zhong, S., & Ghosh, J. (2001). A new formulation of coupled hidden Markov models. Tech. rep., Department of Electrical and Computer Engineering, The University of Texas at Austin.

Zhu, J., Wang, B., Yang, X., Zhang, W., & Tu, Z. (2013). Action recognition with Actons. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3559–3566.