Saying What You're Looking For: Linguistics Meets Video Search

Andrei Barbu, *Student Member, IEEE,* N. Siddharth, *Student Member, IEEE,* and Jeffrey Mark Siskind, *Senior Member, IEEE*

Abstract—We present an approach to searching large video corpora for clips which depict a natural-language query in the form of a sentence. Compositional semantics is used to encode subtle meaning differences lost in other approaches, such as the difference between two sentences which have identical words but entirely different meaning: *The person rode the horse* vs. *The horse rode the person*. Given a sentential query and a natural-language parser, we produce a score indicating how well a video clip depicts that sentence for each clip in a corpus and return a ranked list of clips. Two fundamental problems are addressed simultaneously: detecting and tracking objects, and recognizing whether those tracks depict the query. Because both tracking and object detection are unreliable, our approach uses the sentential query to focus the tracker on the relevant participants and ensures that the resulting tracks are described by the sentential query. While earlier work was limited to single-word queries which correspond to either verbs or nouns, we search for complex queries which contain multiple phrases, such as prepositional phrases, and modifiers, such as adverbs. We demonstrate this approach by searching for 141 sentential queries in 10 Hollywood movies.

Index Terms—Retrieval, video, language, tracking, object detection, event recognition, sentential video retrieval

1 Introduction

Video search engines lag behind text search engines in their wide use and performance. This is in part because the most attractive interface for finding videos remains a natural-language query in the form of a sentence but determining if a sentence describes a video remains a difficult task. This task is difficult for a number of different reasons: unreliable object detectors which are required to determine if nouns occur, unreliable event recognizers which are required to determine if verbs occur, the need to recognize other parts of speech such as adverbs or adjectives, and the need for a representation of the semantics of a sentence which can faithfully encode the desired natural-language query. We propose an approach which simultaneously addresses all of the above problems. Approaches to date generally attempt to independently address the various aspects that make this task difficult. For example, they attempt to separately find videos that depict nouns and videos that depict verbs and essentially take the intersection of these two sets of videos. This general approach of solving these problems piecemeal cannot represent crucial distinctions between otherwise similar input queries. For example, if you search for *The person rode the horse* and for *The horse rode the person*, existing systems would give the same result for both queries as they each contain the same words, but clearly the desired output for these two queries is very different. We develop a holistic approach which both combines tracking and word recognition to address the problems of unreliable object detectors and trackers and at the same time uses compositional semantics to construct the meaning of a sentence from the meaning of its words in order to make crucial but otherwise subtle distinctions between otherwise similar sentences. Given a grammar and an input sentence, we parse that sentence and, for each video clip in a corpus, we simultaneously track all objects that the sentence refers to and enforce the constraint that all tracks must be described by the target sentence using an approach called the *sentence tracker*. Each video is scored by the quality of its tracks, which are guaranteed by construction to depict our target sentence, and the final score correlates with our confidence that the resulting tracks correspond to real objects in the video. We produce a score for every video-sentence pair and return multiple video hits ordered by their scores.

In a recent survey of video retrieval, Hu *et al.* [1] note that work on semantic video search focuses on detecting nouns and verbs, as well as using language to search already-existing video annotation. The state of the art in image retrieval is similar [2]. Note that the approach presented here, by design, would fare poorly on still images as it uses the fact that the input is a video in order to mutually inform and constrain object detection, tracking, and event recognition. Unlike earlier approaches, the work presented here requires no pre-existing annotations aside from a tiny training corpus.

Retrieving clips or frames in which a query object occurs has been addressed both using query-by-example and object detection. Sivic and Zisserman [3] present a statistical local-feature approach to query-by-example. A bounding box is placed around a target object, and frames in which that object occurs are retrieved. Unlike the work presented here, this search is not performed using an object detector, but instead relies on detecting

• *The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907-2035.*
E-mail: andrei@0xab.com, siddharth@iffsid.com, qobi@purdue.edu

Manuscript received November 25, 2013.

regions with similar statistical features. Moreover, it does not exploit the fact that the input is a video, and instead treats each frame of the video independently. Yu *et al.* [4] detect and track a single object, a soccer ball, and recognize actions being performed on that object during a soccer match. They extract gross motion features by examining the position and velocity of the object in order to recognize events and support a small number of domain-specific actions limited to that specific single object. Anjulan and Canagarajah [5] track stable image patches to extract object tracks over the duration of a video and group similar tracks into object classes. Without employing an object detector, these methods cannot search a collection of videos for a particular object class but instead must search by example. Byrne *et al.* [6] employ statistical local features, such as Gabor features, to perform object detection. These do not perform as well as more recent object detectors on standard benchmarks such as PASCAL VOC. Sadeghi and Farhadi [7] recognize objects, in images, in the context of their spatial relations, using an object detector. They train an object detector not just for an object class, but for a combination of multiple interacting objects. This allows them to detect more complex scenarios, such as a person riding a horse, by building targeted object detectors. Moreover, knowledge of the target scenario improves the performance of the object detector. Similarly, in our work, knowledge about the query improves the performance of each of the individual detectors for each of the words in the query. But their approach differs fundamentally from the one presented here because it is not compositional in nature. In order to detect *The person rode the horse*, one must train on examples of exactly that entire sentence, whereas in the work presented here, independent detectors for *person*, *horse*, and *ride* combine together to encode the semantics of the sentence and to perform retrieval of a sentence that may never have occurred in the training set.

Prior work on verb detection does not integrate with work on object detection. Chang *et al.* [10] find one of four different highlights in basketball games using hidden Markov models and the expected structure of a basketball game. They do not detect objects but instead classify entire presegmented clips, are restricted to a small number of domain-specific actions, and support only single-word queries. Event recognition is a popular subarea of computer vision but has remained limited to single-word queries [11], [12], [13], [14], [15]. We will avail ourselves of such work later [16] to show that the work presented here both allows for richer queries and improves on the performance of earlier approaches.

Prior work on more complex queries involving both nouns and verbs essentially encodes the meaning of a sentence as a conjunction of words, largely discarding the compositional semantics of the sentence reflected by sentence structure. Christel *et al.* [17], Worring *et al.* [18], and Snoek *et al.* [19] present various combinations of text search, verb retrieval, and noun retrieval, and essentially allow for finding videos which are at the intersection of multiple search mechanisms. Aytar *et al.* [20] rely on annotating a video corpus with sentences that describe each video in that corpus. They employ text-based search methods which given a query, a conjunction of words, attempt to find videos of similar concepts as defined by the combination of an ontology and statistical features of the videos. Their model for a sentence is a conjunction of words where higher-scoring videos more faithfully depict each individual word but the relationship between words is lost. None of these methods attempt to faithfully encode the semantics of a sentence and none of them can encode the distinction between *The person hit the ball* and *The ball hit the person*.

In what follows, we describe a system, which unlike previous approaches, allows for a natural-language query of video corpora which have no human-provided annotation. Given a sentence and a video corpus, we retrieve a ranked list of videos which are described by that sentence. We show a method for constructing a lexicon with a small number of parameters, which are reused among multiple words, making training those parameters easy and ensuring the system need not be shown positive examples of every word in the lexicon. We present a method for combining models for individual words into a model for an entire sentence and for recognizing that sentence while simultaneously tracking objects in order to score a video-sentence pair. To demonstrate this approach, we run 141 natural-language queries on a corpus of 10 full-length Hollywood movies using a grammar which includes nouns, verbs, adjectives, adverbs, spatial-relation prepositions, and motion prepositions. This is the first approach which can search for complex queries which include multiple phrases, such as prepositional phrases, and modifiers, such as adverbs.

2 TRACKING

We begin by describing the operation of a detection-based tracker on top of which the sentence tracker will be constructed. To search for videos which depict a sentence, we must first track objects that participate in the event described by that sentence. Tracks consist of a single detection per frame per object. To recover these tracks, we employ detection-based tracking. An object detector is run on every frame of a video, producing a set of axis-aligned rectangles along with scores which correspond to the strength of each detection. We employ the Felzenszwalb *et al.* [21], [22] object detector, specifically the variant developed by Song *et al.* [23]. There are two reasons why we need a tracker and cannot just take the top-scoring detection in every frame. First, there may be multiple instances of the same object in the field of view. Second, object detectors are extremely unreliable. Even on standard benchmarks, such as the PASCAL Visual Object Classes (VOC) Challenge, even the best detectors for the easiest-to-detect object classes achieve average-precision scores of 40% to 50% [24]. We overcome both of these problems by integrating the intra-frame information available from the object detector with inter-

frame information computed from optical flow.

We expect that the motion of correct tracks agrees with the motion of the objects in the video which we can compute separately and independently of any detections using optical flow. We call this quantity the motion coherence of a track. In other words, given a detection corresponding to an object in the video, we compute the average optical flow inside that detection, forward-project the detection along that vector, and expect to find a strong detection in the next frame at that location. We formalize this intuition into an algorithm which finds an optimal track given a set of detections in each frame. For each frame t in a video of length T, each detection j has an associated axis-aligned rectangle b_j^t and score $f(b_j^t)$ and each pair of detections in adjacent frames has an associated motion coherence score $g(b_{j^{t-1}}^{t-1}, b_{j^t}^t)$. We formulate the score of a track $\mathbf{j} = \langle j^1, \ldots, j^T \rangle$ as

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

where we take g, the motion coherence, to be a nonincreasing function of the squared Euclidean distance between the center of $b_{j^{t-1}}^{t-1}$ and the center of $b_{j^t}^t$ projected one frame forward. While the number of possible tracks is exponential in the number of frames in the video, Eq. 1 can be maximized in time linear in the number of frames and quadratic in the number of detections per frame using dynamic programming [25], the Viterbi [26], [27] algorithm.

The development of this tracker follows that of Barbu *et al.* [28] which presents additional details of such a tracker, including an extension which allows generating multiple tracks per object class using non-maxima suppression. That earlier tracker used the raw detection scores from the Felzenszwalb *et al.* [21], [22] object detector. These scores are difficult to interpret because the mean and variance of scores varies by object class making it difficult to decide whether a detection is strong. To get around this problem, we pass all detections through a sigmoid $\frac{1}{1+\exp(-b(t-a))}$ whose center, a, is the model threshold and whose scaling factor b, is 2. This normalizes the score to the range $[0, 1]$ and makes scores more comparable across models. In addition, the motion coherence score is also passed through a similar sigmoid, with center 50 and scale $-1/11$.

3 WORD RECOGNITION

Given tracks, we want to decide if a word describes one or more of those tracks. This is a generalization of event recognition, generalizing the notion of an event from verbs to other parts of speech. To recognize if a word describes a collection of tracks, we extract features from those tracks and use those features to formulate the semantics of words. Word semantics are formulated in terms of finite state machines (FSMs) which accept one or more tracks. Fig. 2 provides an overview of the FSMs used in Sections 6.2 and 6.3, rendered as regular expressions. This approach is a limiting case of that taken by Barbu *et al.* [29] which used hidden Markov models (HMMs) to encode the semantics of verbs. In essence, our FSMs are unnormalized HMMs with binary transition matrices and binary output distributions. This allows the same recognition mechanism as that used by Barbu *et al.* [29] to be employed here.

We construct word meanings in two levels. First, we construct 18 predicates, shown in Fig. 1, which accept one or more detections. We then construct word meanings for our lexicon of 15 words, shown in Fig. 2, as regular expressions which accept tracks and are composed out of these predicates. This two-level construction allows sharing low-level features and parameters across words. All words share the same predicates which are encoded relative to 9 parameters: **far**, **close**, **stationary**, **Δclosing**, **Δangle**, **Δpp**, **Δquickly**, **Δslowly**, and **overlap**. These parameters are learned from a tiny number of positive and negative examples that cover only a fraction of the words in the lexicon. To make predicates independent of the video resolution, detections are first rescaled relative to a standard resolution of 1280×720, otherwise parameters such as **far** would need to vary with resolution.

Given a regular expression for a word, we can construct a nondeterministic FSM, with one accepting state, whose allowable transitions are encoded by a binary transition matrix h, giving score zero to allowed transitions and $-\infty$ to disallowed transitions, and whose states accept detections which agree with the predicate a, again with the same score of zero or $-\infty$. With this FSM, we can recognize if a word describes a track $\langle \hat{j}^1, \ldots, \hat{j}^T \rangle$, by finding

$$\max_{k^1,\ldots,k^T} \sum_{t=1}^{T} h(k^t, b_{\hat{j}^t}^t) + \sum_{t=2}^{T} a(k^{t-1}, k^t) \tag{2}$$

where k^1 through k^{T-1} range over the set of states of the FSM and k^T is the singleton set containing the accepting state. If this word describes the track, the score yielded by Eq. 2 will be zero. If it does not, the score will be $-\infty$. The above formulation can be generalized to multiple tracks and is the same as that used by Barbu *et al.* [28]. We find accepting paths through the lattice of states again using dynamic programming, the Viterbi algorithm. Note that this method can be applied to encode not just the meaning of verbs but also of other parts of speech. For example, the meaning of a static concept, such as a preposition like *left-of* that encodes a temporally invariant spatial relation, can be encoded as a single-state FSM whose output predicate encodes that relation. The meaning of a dynamic concept, such as a preposition like *towards* that encodes temporally variant motion, can be encoded in a multi-state FSM much like a verb. It is well known in linguistics that the correspondence between semantic classes and parts of speech is flexible. For example, some verbs, like *hold*, encode static concepts, while some nouns, like *wedding*, encode dynamic concepts. Employing a uniform but powerful representation to encode the meaning of

all parts of speech supports this linguistic generality and further allows a single but powerful mechanism to build up the semantics of sentences from the semantics of words. This same general mechanism admits some resiliency to noisy input by allowing one to construct FSMs with 'garbage' states that accept noisy segments. We avail ourselves of this capacity by incorporating $\mathbf{true}^+$ into many of the word FSMs in Fig. 2.

4 SENTENCE TRACKER

Our ultimate goal is to search for videos described by a natural-language query in the form of a sentence. The framework developed so far falls short of supporting this goal in two ways. First, as we attempt to recognize multiple words that constrain a single track, it becomes unlikely that the tracker will happen to produce an optimal track which satisfies all the desired predicates. For example, when searching for a person that is both *running* and doing so *leftward*, the chance that there may be a single noisy frame that fails to satisfy either the *running* predicate or the *leftward* predicate is greater than for a single-word query. Second, a sentence is not a conjunction of words, even though a word is represented here as a conjunction of features, so a new mechanism is required to faithfully encode the compositional semantics of a sentence as reflected in its structure. Intuitively, we must encode the mutual dependence in the sentence *The tall person rode the horse* so that the person is tall, not the horse, and the person is riding the horse, not vice versa.

We address the first point by biasing the tracker to produce tracks which agree with the predicates that are being enforced. This may result in the tracker producing tracks which have to consist of lower-scoring detections, which decreases the probability that these tracks correspond to real objects in the video, This is not a concern as we will present the users with results ranked by their tracker score. In essence, we pay a penalty for forcing a track to agree with the enforced predicates and the ultimate rank order is influenced by this penalty. The computational mechanism that enables this exists by virtue of the fact that our tracker and word recognizer have the same internal representation and algorithm, namely, each finds optimal paths through a lattice of scored detections, $f(b^t_{j^t})$, for the tracker, or states scored by their output predicate, $h(k^t, b^t_{j^t})$, for the word recognizer, and each weights the links in that lattice by a score, the motion coherence, $g(b^{t-1}_{j^{t-1}}, b^t_{j^t})$, for the tracker, and state-transition score, $a(k^{t-1}, k^t)$, for the word recognizer. We simultaneously find the track $j^1, \ldots, j^T$ and state sequence $k^1, \ldots, k^T$ that optimizes a joint objective function

$$\max_{j^1,\ldots,j^T} \max_{k^1,\ldots,k^T} \left(\sum_{t=1}^{T} f(b^t_{j^t}) + \sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}}, b^t_{j^t}) + \sum_{t=1}^{T} h(k^t, b^t_{j^t}) + \sum_{t=2}^{T} a(k^{t-1}, k^t) \right) \quad (3)$$

which ensures that, unless the state sequence for the word FSM leads to an accepting state, the resulting aggregate score will be $-\infty$. This constrains the track to depict the word and finds the highest-scoring one that does so. Intuitively, we have two lattices, a tracker lattice and a word-recognizer lattice, and we find the optimal path, again with the Viterbi algorithm, through the cross-product of these two lattices. This cross-product lattice construction is shown in Fig. 3.

The above handles only a single word, but given a sentential query we want to encode its semantics in terms of multiple words and multiple trackers. We parse an input sentence with a grammar, shown in Fig. 5, and extract the number of participants and the track-to-role mapping. Each sentence that describes an event has a number of roles that must be filled with entities that serve as participants in that event. For example, in the sentence *The person rode the horse quickly away from the other horse*, there are three participants, one person and two horses, and each of the three participants plays a different role in the sentence, *agent* (the entity performing the action, in this case the person), *patient* (the entity affected by the action, in this case the first horse), and *goal* (the destination of the action, in this case the second horse). Each word in this sentence refers to a subset of these three different participants, as shown in Fig. 4, and words that refer to multiple participants, such as *ride*, must be assigned participants in the correct argument order to ensure that we encode *The person rode the horse* rather than *The horse rode the person*. We use a custom natural-language parser which takes as input a grammar, along with the arity and thematic roles of each word, and computes a track-to-role mapping: which participants fill which roles in which words. We employ the same mechanism as described above for simultaneous word recognition and tracking, except that we instantiate one tracker for each participant and one word recognizer for each word. The thematic roles, θ^n_w, map the nth role in a word w to a tracker. Fig. 4 displays an overview of this mapping for a sample sentence. Trackers are shown in red, word recognizers are shown in blue, and the track-to-role mapping is shown using the arrows. Given a sentential query that has W words, L participants, and track-to-role mapping θ^n_w, we find a collection of tracks $\langle j^1_1, \ldots, j^T_1 \rangle, \ldots, \langle j^1_L, \ldots, j^T_L \rangle$, one for each participant, and accepting state sequences $\langle k^1_1, \ldots, k^T_1 \rangle, \ldots, \langle k^1_W, \ldots, k^T_W \rangle$, one for each word, that optimizes a joint objective function

$$\max_{\substack{j^1_1,\ldots,j^T_1 \\ \vdots \\ j^1_L,\ldots,j^T_L}} \max_{\substack{k^1_1,\ldots,k^T_1 \\ \vdots \\ k^1_W,\ldots,k^T_W}} \left(\sum_{l=1}^{L} \sum_{t=1}^{T} f(b^t_{j^t_l}) + \sum_{t=2}^{T} g(b^{t-1}_{j^{t-1}_l}, b^t_{j^t_l}) + \sum_{w=1}^{W} \sum_{t=1}^{T} h_w(k^t_w, b^t_{j^t_{\theta^1_w}}, b^t_{j^t_{\theta^2_w}}) + \sum_{t=2}^{T} a_w(k^{t-1}_w, k^t_w) \right) \quad (4)$$

where a_w and h_w are the transition matrices and predicates for word w, $b^t_{j^t_l}$ is a detection in the tth frame

$$
\begin{array}{ll}
\text{FAR}(a,b) & \triangleq |a_{cx} - b_{cx}| - \frac{a_{width}}{2} - \frac{b_{width}}{2} > \textbf{far} \\
\text{REALLY-CLOSE}(a,b) & \triangleq |a_{cx} - b_{cx}| - \frac{a_{width}}{2} - \frac{b_{width}}{2} > \frac{\textbf{close}}{2} \\
\text{CLOSE}(a,b) & \triangleq |a_{cx} - b_{cx}| - \frac{a_{width}}{2} - \frac{b_{width}}{2} > \frac{\textbf{close}}{2} \\
\text{STATIONARY}(b) & \triangleq \text{flow-magnitude}(b) \leq \textbf{stationary} \\
\text{CLOSING}(a,b) & \triangleq |a_{cx} - b_{cx}| > |\text{project}(a)_{cx} - \text{project}(b)_{cx}| + \Delta\textbf{closing} \\
\text{DEPARTING}(a,b) & \triangleq |a_{cx} - b_{cx}| < |\text{project}(a)_{cx} - \text{project}(b)_{cx}| + \Delta\textbf{closing} \\
\text{MOVING-DIRECTION}(a,b,\alpha) & \triangleq |\text{flow-orientation}(a) - \alpha|^\circ < \Delta\textbf{angle} \ \wedge \\
 & \quad \text{flow-magnitude}(a) > \textbf{stationary} \\
\text{LEFT-OF}(a,b) & \triangleq a_{cx} < b_{cx} + \Delta\textbf{pp} \\
\text{RIGHT-OF}(a,b) & \triangleq a_{cx} > b_{cx} + \Delta\textbf{pp} \\
\text{LEFTWARD}(a,b) & \triangleq \text{MOVING-DIRECTION}(a,b,0) \\
\text{LEFTWARD}(a,b) & \triangleq \text{MOVING-DIRECTION}(a,b,\pi) \\
\text{STATIONARY-BUT-FAR}(a,b) & \triangleq \text{FAR}(a,b) \wedge \text{STATIONARY}(a) \wedge \text{STATIONARY}(b) \\
\text{STATIONARY-BUT-CLOSE}(a,b) & \triangleq \text{CLOSE}(a,b) \wedge \text{STATIONARY}(a) \wedge \text{STATIONARY}(b) \\
\text{MOVING-TOGETHER}(a,b) & \triangleq |\text{flow-orientation}(a) - \text{flow-orientation}(b)|^\circ < \Delta\textbf{angle} \wedge \\
 & \quad \text{flow-magnitude}(a) > \textbf{stationary} \wedge \\
 & \quad \text{flow-magnitude}(b) > \textbf{stationary} \\
\text{APPROACHING}(a,b) & \triangleq \text{CLOSING}(a,b) \wedge \text{STATIONARY}(b) \\
\text{QUICKLY}(a) & \triangleq \text{flow-magnitude}(a) > \Delta\textbf{quickly} \\
\text{SLOWLY}(a) & \triangleq \textbf{stationary} < \text{flow-magnitude}(a) < \Delta\textbf{slowly} \\
\text{OVERLAPPING}(a,b) & \triangleq \frac{a \cap b}{a \cup b} \geq \textbf{overlap}
\end{array}
$$

Fig. 1. Predicates which accept detections, denoted by a and b, formulated around 9 parameters. These predicates are used for the second and third experiment, Sections 6.2 and 6.3. Predicates for the first experiment, Section 6.1, are similar and provided in the appendix. The function project projects a detection forward one frame using optical flow. The functions flow-orientation and flow-magnitude compute the angle and magnitude of the average optical-flow vector inside a detection. The function a_{cx} accesses the x coordinate of the center of a detection. The function a_{width} computes the width of a detection. The functions $\cup$ and $\cap$ compute the area of the union and intersection of two detections respectively. The function $|\cdot|^\circ$ computes angular separation. Words are formed as regular expressions over these predicates.

of the lth track, and $b^t_{j^t_{\theta^n_w}}$ connects a participant that fills the nth role in word w with the detections of its tracker. Since the aggregate score will be $-\infty$ if even a single word-recognizer score would be $-\infty$, this equation constrains the subcollection of tracks that play roles in each of the words in the sentence to satisfy the semantic conditions for that word, collectively constraining the entire collection of tracks for all of the participants to satisfy the semantic conditions for the entire sentence. Further, it finds that collection of tracks with maximal tracker score sum. In essence, for each word, we take the cross product of its word lattice with all of the tracker lattices that fill roles in that word, collectively taking a single large cross product of all word and tracker lattices in a way that agrees with the track-to-role mapping, and find the optimal path through the resulting lattice. This allows us to employ the same computational mechanism, the Viterbi algorithm, to find this optimal node sequence. The resulting tracks will satisfy the semantics of the input sentence, even if this incurs a penalty by having to choose lower-scoring detections.

5 Retrieval

We employ the mechanisms developed above to perform video retrieval given a sentential query. Given a corpus of videos, we retrieve short clips which depict a full sentence from these longer videos. To do so, we use the fact that the sentence tracker developed above scores a video-sentence pair. The sentence-tracker score sums the scores of the participant trackers and the scores of the word recognizers. As explained in the previous section, the word-recognizer score, and thus the sum of all such, is either 0 or $-\infty$. This means that the aggregate sentence-tracker score will be $-\infty$ if if no tracks can be found which depict the query sentence. Otherwise, it will simply be the tracker-score sum. This score indicates our confidence in how well a video depicts a query sentence, the better the tracker score the more confident we can be that the tracks correspond to real objects in the video. The fact that those tracks are produced at all ensures that they depict the query sentence. We take this correlation between score and whether a video depicts a sentence to perform video retrieval. Given a corpus of clips, we run

$$
\begin{aligned}
\texttt{horse}(a) &\triangleq (a_{\text{object-class}} = \text{“horse”})^{+} \\
\texttt{person}(a) &\triangleq (a_{\text{object-class}} = \text{“person”})^{+} \\
\texttt{quickly}(a) &\triangleq \textbf{true}^{+}\ \text{QUICKLY}(a)^{\{3,\}}\ \textbf{true}^{+} \\
\texttt{slowly}(a) &\triangleq \textbf{true}^{+}\ \text{SLOWLY}(a)^{\{3,\}}\ \textbf{true}^{+} \\
\texttt{from the left}(a,b) &\triangleq \textbf{true}^{+}\ \text{LEFT-OF}(a,b)^{\{5,\}}\ \textbf{true}^{+} \\
\texttt{from the right}(a,b) &\triangleq \textbf{true}^{+}\ \text{RIGHT-OF}(a,b)^{\{5,\}}\ \textbf{true}^{+} \\
\texttt{leftward}(a) &\triangleq \textbf{true}^{+}\ \text{LEFTWARD}(a)^{\{5,\}}\ \textbf{true}^{+} \\
\texttt{rightward}(a) &\triangleq \textbf{true}^{+}\ \text{RIGHTWARD}(a)^{\{5,\}}\ \textbf{true}^{+} \\
\texttt{to the left of}(a,b) &\triangleq \textbf{true}^{+}\ \text{LEFT-OF}(a,b)^{\{3,\}}\ \textbf{true}^{+} \\
\texttt{to the right of}(a,b) &\triangleq \textbf{true}^{+}\ \text{RIGHT-OF}(a,b)^{\{3,\}}\ \textbf{true}^{+} \\
\texttt{towards}(a,b) &\triangleq \text{STATIONARY-BUT-FAR}(a,b)^{+}\ \text{APPROACHING}(a,b)^{\{3,\}} \\
&\quad\ \text{STATIONARY-BUT-CLOSE}(a,b)^{+} \\
\texttt{away from}(a,b) &\triangleq \text{STATIONARY-BUT-CLOSE}(a,b)^{+}\ \text{DEPARTING}(a,b)^{\{3,\}} \\
&\quad\ \text{STATIONARY-BUT-FAR}(a,b)^{+} \\
\texttt{ride}(a,b) &\triangleq \textbf{true}^{+}\ (\text{MOVING-TOGETHER}(a,b) \wedge \text{OVERLAPPING}(a,b))^{\{5,\}}\ \textbf{true}^{+} \\
\texttt{lead}(a,b) &\triangleq \textbf{true}^{+} \left(\begin{array}{l} \neg\text{REALLY-CLOSE}(a,b) \wedge \\ \text{MOVING-TOGETHER}(a,b) \wedge \\ \left(\begin{array}{l} (\text{LEFT-OF}(a,b) \wedge \text{LEFTWARD}(a)) \vee \\ (\text{RIGHT-OF}(a,b) \wedge \text{RIGHTWARD}(a)) \end{array} \right) \end{array} \right)^{\{5,\}} \textbf{true}^{+} \\
\texttt{approach}(a,b) &\triangleq \textbf{true}^{+}\ \text{APPROACHING}(a,b)^{\{5,\}}\ \textbf{true}^{+}
\end{aligned}
$$

Fig. 2. Regular expressions which encode the meanings of each of the 15 words or lexicalized phrases in the lexicon used for the second and third experiment, Sections 6.2 and 6.3. These are composed from the predicates shown in Fig. 1. Regular expressions for the first experiment, Section 6.1, are similar and provided in the appendix. We use an extended regular-expression syntax where an exponent of $\{t,\}$ allows a predicate to hold for t or more frames.

the sentence tracker with the query sentence on each clip. Clips are then ranked by their sentence-tracker score.

The above approach retrieves short clips from a corpus of such. Our ultimate goal, however, is to take, as input, videos of arbitrary length and find short clips which depict the query sentence from these longer videos. The sentence tracker is able to find a single instance of an event in a long video because, as shown in Fig. 2, word meanings have garbage states of unbounded length prepended and appended to them. But this would produce a single detected event for each long video instead of potentially many short clips for each input video. To produce multiple clips, we split all input videos into short, several second long, clips and produce a corpus of clips on which we perform video retrieval. The exact clip length is unimportant as long as the query sentences can be fully depicted in the clip length because, as noted above, the sentence tracker will find shorter events in a longer clip. This also motivates the use of fixed-length clips as all words in our chosen lexicon depict short events. One downside of this is the inability to detect events that straddle clip boundaries. To address this problem, we segment input videos into short but overlapping clips, ensuring that each clip boundary is contained within another clip.

Given the corpus of clips to be searched, the other piece of information required is the query sentence. The sentence is first parsed according to the grammar shown in Fig. 5. The grammar presented is context-free and the sentence is parsed using a standard recursive-descent parser. Note that the grammar presented here is infinitely recursive. Noun phrases optionally contain prepositional phrases which contain other noun phrases. For one example one might say: *The person to the left of the horse to the right of the person to the left of the horse* .... The words shown in Fig. 2 require arguments and each of these arguments has one of five thematic roles: *agent*, *patient*, *referent*, *goal*, and *source*. The parse tree, together with the role information, are used to determine the number of participants and which participants fill which roles in the event described by the sentence. This provides the track-to-role mapping, θ, in Eq. 4.

An alternate method for producing this mapping would be to employ a more general natural-language parser such as the Stanford Parser [30]. Given an input sentence such as *The person rode the horse toward the horse*, the Stanford Parser produces the following dependencies

```
det(person-2, The-1)
nsubj(rode-3, person-2)
root(ROOT-0, rode-3)
det(horse-5, the-4)
dobj(rode-3, horse-5)
det(horse-8, the-7)
prep_toward(rode-3, horse-8)
```

which can also be used to construct the requisite track-

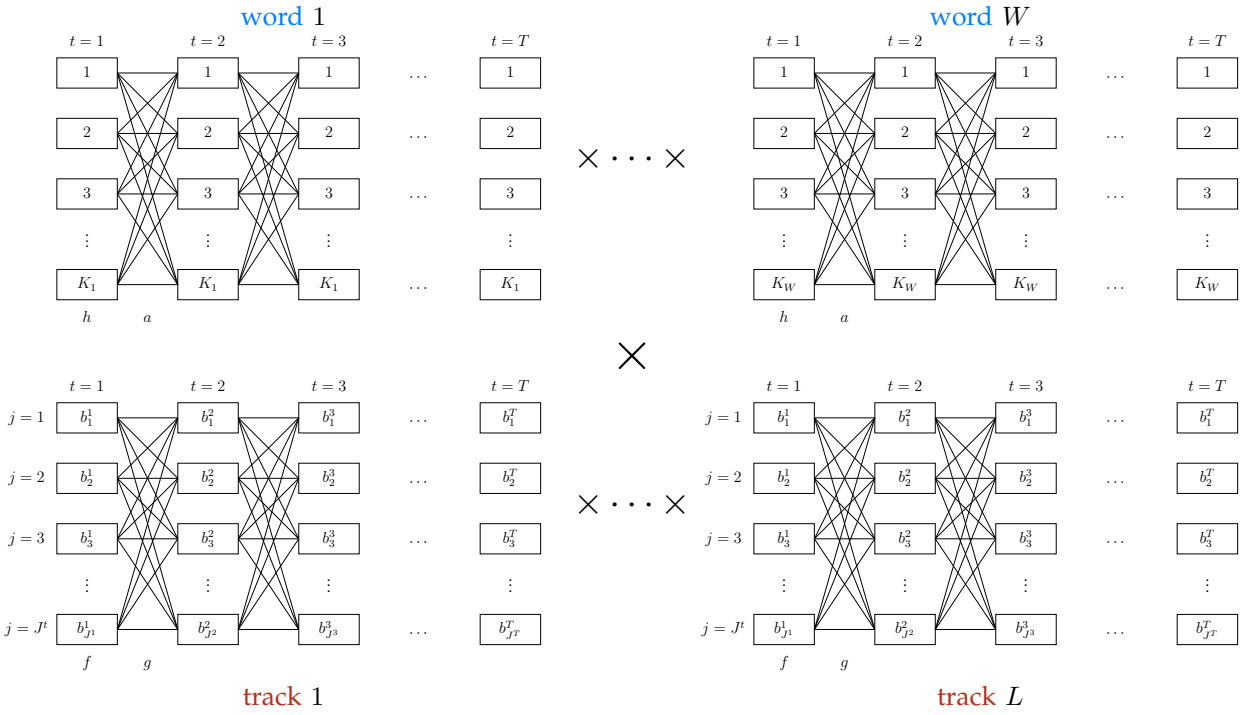


Fig. 3. Tracker lattices are used to track each participant. Word lattices constructed from word FSMs for each word in the sentence recognize collections of tracks for participants that exhibit the semantics of that word as encoded in the FSM. We take the cross product of multiple tracker and word lattices to simultaneously track participants and recognize words. This ensures that the resulting tracks are described by the desired sentence.

to-role mapping. The output above correctly identifies three participants, *person-2*, *horse-5*, and *horse-8*. Note how the transitive verb *rode-3* distinguishes between its two arguments, identifying *person-2* as its subject and *horse-5* as its direct object. Using a general natural-language parser would allow a retrieval system to handle a much larger space of sentences and alleviate the need to specify the grammar and track-to-role mapping mechanism for each word. This approach would still require specification of the semantics of each word. We construct the exposition and experiments around the first approach, with a custom grammar and parser, to render the algorithm and its requirements more transparent.

The above procedure for searching a corpus of clips can be sped up significantly when searching the same corpus with multiple sentential queries. First, the object detections required for the sentence tracker are independent of the query sentence. In other words, the object detector portion of the lattice, namely the score, position, and optical flow for each detection, are unaffected by the query sentence even though the tracks produced are affected by it. This can be seen in Eq. 4 where neither f (the detection score), g (the motion coherence), nor either of their arguments depend on k (the lexical entry of a word), or θ (the track to role mapping). This allows us to preprocess the video corpus and compute object detections and optical-flow estimates which can be reused with different sentential queries. This constitutes the majority of the runtime of the algorithm; object detection and optical-flow estimation are an order of magnitude slower than parsing and sentence-tracker inference.

The first speedup addressed how to decrease the computation for each clip in the corpus. The second addresses the fact that the resulting retrieval algorithm still requires inspecting every clip in the corpus to determine if it depicts the query sentence. We ameliorate this problem by first noting that the lexicon and grammar presented in Figs. 2 and 5 have no negation. This means that in order for a video to depict a sentence it must also depict any fragment of that sentence. By sentence fragment, we mean any subsequence of a word string that can be generated by any terminal or nonterminal in the grammar. For example, the sentence *The person approached the horse quickly* has sentence fragments *person*, *horse*, *approached*, *approached the horse*, *quickly*, and *approached the horse quickly* Any video depicting this entire sentence must also depict these fragments. Were our grammar to have negation, this would not be true; a video depicting the sentence *The person did not approach the horse* would not depict the fragment *approach the horse*. This leads to an efficient algorithm for reusing earlier queries to speed up novel queries. Intuitively, if you've already determined that nothing approaches a horse in a clip, nothing will approach a horse quickly in that clip. In other words, one can parse the query sentence and look through all previous queries, potentially queries of sentence fragments, to see which queries form subtrees of the current query. All clips which have score $-\infty$ for these

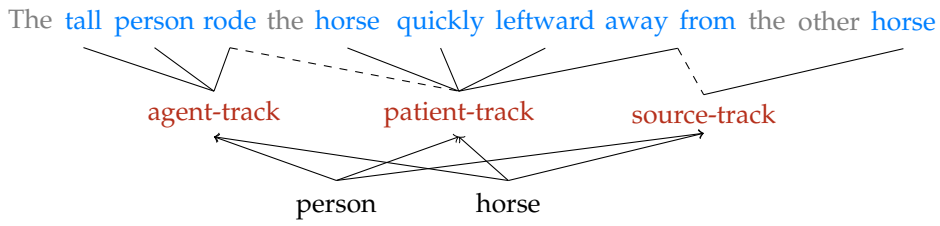


Fig. 4. Different sentential queries lead to different cross products. The sentence is parsed and the role of each participant, show in red, is determined. A single tracker lattice is constructed for each participant. Words and lexicalized phrases, shown in blue, have associated word lattices which encode their semantics. The arrows between words and participants represent the track-to-role mappings, θ, required to link the tracker and word lattices in a way that faithfully encodes the sentential semantics. Some words, like determiners, shown in grey, have no semantics beyond determining the parse tree and track-to-role mapping. The dashed lines indicate that the argument order is essential for words which have more than one role. In other words, predicates like *ride* and *away from* are not symmetric. Detection sources are shown in black, in this case two object detectors. The tracker associated with each participant has access to all detection sources, hence the bipartite clique between the trackers and the detection sources.

S	→	NP VP	NP	→	D [A] N [PP]
D	→	*an* | *the*	A	→	*blue* | *red*
N	→	*person* | *horse* | *backpack* | *trash can* | *chair* | *object*	PP	→	P NP
P	→	*to the left of* | *to the right of*	VP	→	V NP [Adv] [PP_M]
V	→	*approached* | *lead* | *carried* | *picked up* | *put down* | *rode*	Adv	→	*quickly* | *slowly*
PP_M	→	P_M NP | *from the left* | *from the right*	P_M	→	*towards* | *away from*

Fig. 5. The grammar for sentential queries used in Section 6. Items in black are shared between all experiments. Items in red are exclusive to the first experiment, Section 6.1. Items in blue are exclusive to the second and third experiments, Sections 6.2 and 6.3.

shorter queries can be eliminated from consideration when searching for the longer query. This enables scaling to much larger video corpora by immediately eliminating videos which cannot depict the query sentence.

6 Results

We present three experiments which test video retrieval using sentential queries. The first employs a corpus of clips collected specifically for this task which facilitates more complex queries with a larger number of nouns and participants while being designed to stress the system by employing a multitude of nearly-identical query sentences. The second and third employ a corpus of 10 full-length Hollywood movies showing the ability of this approach to handle videos found in the wild and not filmed specifically for this task.

6.1 The new^3 corpus

We first evaluate this approach on a corpus[1] (called new^3) of 94 short clips shot outdoors from a stationary camera. These clips show between one and two people performing actions with one or two objects, selected from a collection of three objects, all present in the field of view in every clip. The language fragment supported by this corpus includes five nouns: one for each of the objects, one for people, and a generic noun *object*. The later experiments on Hollywood movies only support two nouns: *person* and *horse*, due to the fact that object detector performance is much lower on this more challenging set of videos. Frames from sample videos in new^3 are shown in Fig. 7. To search this corpus using sentential queries we first formulate the semantics of a small fragment of English consisting of 17 lexical items (5 nouns, 2 adjectives, 4 verbs, 2 adverbs, 2 spatial-relation prepositions, and 2 motion prepositions). The grammar and lexicon for this fragment of English are presented in Fig. 5 and consist of the black and red portions of the figure. The semantics of the words are formed similarly to Figs. 1 and 2 and the corresponding figures for this experiment are provided in the appendix. We form 21 query sentences, shown in Fig. 6, including in these all possible minimal pairs which can be constructed using the given grammar while ignoring the infinite recursion in the noun phrase. By a minimal pair we mean two sentences which differ only in one part of speech. For example the sentences

The red object approached the chair.
The blue object approached the chair.

form a minimal pair which differs only in the adjective in the subject noun phrase. This ensures a more difficult test where each part of speech in each query must be correctly interpreted.

1. The new^3 video corpus along with human annotations is available at http://0xab.com/research/video-events/new3-corpus.tar.gz.

We evaluate this approach by running each sentential query against the video corpus. Chance performance, the probability that a sentence is depicted by a randomly selected video, is 13.12%. Given a sentential query, the top-scoring video for that sentence depicts that sentence 85.71% of the time. This shows that we are able to successfully retrieve clips given sentential queries even when many queries form minimal pairs which are difficult to distinguish and even when we restrict ourselves to only the top hit.

6.2 Ten westerns

We further demonstrate the utility of this approach on a more challenging corpus composed of 10 Hollywood westerns: Black Beauty (Warner Brothers, 1994), The Black Stallion (MGM, 1979), Blazing Saddles (Warner Brothers, 1974), Easy Rider (Columbia Pictures, 1969), The Good the Bad and the Ugly (Columbia Pictures, 1966), Hidalgo (Touchstone Pictures, 2004), National Velvet (MGM, 1944), Once Upon a Time in Mexico (Columbia Pictures, 2003), Seabiscuit (Universal Pictures, 2003), and Unforgiven (Warner Brothers, 1992). In total, this video corpus has 1187 minutes of video, roughly 20 hours. We temporally downsampled all videos to 6 frames per second but kept their original spatial resolutions which varied from 336×256 pixels to 1280×544 pixels with a mean resolution of 659.2×332.8 pixels. We split these videos into 37187 clips, each clip being 18 frames (3 seconds) long, while overlapping the previous clip by 6 frames. This overlap ensures that actions that might otherwise occur on clip boundaries will also occur as part of a clip. While there is prior work on shot segmentation [31] we did not employ it for two reasons. First, it complicates the system and provides an avenue for additional failure modes. Second, the approach taken here is able to find an event inside a longer video with multiple events. The only reason why we split the videos into clips is to return multiple hits.

We adopt the grammar from Fig. 5, specifically the black and blue portions. This grammar allows for sentences that describe people interacting with horses, hence our choice of genre for the video corpus, namely westerns. A requirement for determining whether a video depicts a sentence, and the degree to which it depicts that sentence, is to detect the objects that might fill roles in that sentence. Previous work has shown that people and horses are among the easiest-to-detect objects, although the performance of object detectors, even for these classes, remains extremely low. To ensure that we did not test on the training data, we employed previously-trained object models that have not been trained on these videos but have instead been trained on the PASCAL VOC Challenge [24]. We use models trained by the UoCTTI_LSVM-MDPM team (the authors of Felzenszwalb *et al.* [21], [22]) for the 2009 Challenge. On the 2009 Challenge, the *person* model achieves an AP score of 41.5% and the *horse* model achieves an AP score of 38.0%. We note that the improvement in AP scores for these object classes in subsequent years of the Challenge has been minor. We also require settings for the 9 parameters, shown in Fig. 1, which are required to produce the predicates which encode the semantics of the words in this grammar. We trained all 9 parameters simultaneously on only 3 positive examples and 3 negative examples. Note that these training examples cover only a subset of the words in the grammar but are sufficient to define the semantics of all words because this word subset touches upon all the underlying parameters. Training proceeded by exhaustively searching a small uniform grid, with between 3 and 10 steps per dimension, of all nine parameter settings to find a combination which best classified all 6 training samples which were then removed from the test set. Yu and Siskind *et al.* [32] present an alternate strategy for training the parameters of a lexicon of words given a video corpus.

We generated 204 sentences that conform to the grammar in Fig. 5 from the template shown in Fig. 8. We eliminated the 63 queries that involve people riding people and horses riding people or other horses, as our video corpus has no positive examples for these sentences. This leaves us with 141 queries which conform to our grammar. For each sentence, we scored every clip paired with that sentence and return the top 10 best-scoring clips for that sentence. Each of these top 10 clips was annotated by a human judge with a binary decision: is this sentence true of this clip? In Fig. 10(a), we show the average precision of the system over all 141 queries on the top 10 hits for each query as a function of a threshold on the scores. As the threshold nears zero, the system may return fewer than 10 results per sentence because it eliminates query results which are unlikely to be true positives. As the threshold tends to $-\infty$, the average precision across all top 10 clips for all sentences is 22.9%, and at its peak, the average precision is 72.4%. In Fig. 10(b), we show the number of results returned per sentence, eliminating those results which have a score of $-\infty$ since that means that no tracks could be found which agree with the semantics of the sentence, On average, there are 7.96 hits per sentence, with standard deviation 3.61, and with only 14 sentences having no hits. In Fig. 10(c), we show the number of correct hits per sentence. On average, there are 1.83 correct hits per sentence, with standard deviation 2.26, and with 80 sentences having at least one true positive.

We highlight the usefulness of this approach in Fig. 11 where we show the top 6 hits for two similar queries: *The person approached the horse* and *The horse approached the person*.[2] Hits are presented in order of score, with the highest scoring hit at the top and scores decreasing as one moves down. Note how the results for the two sentences are very different from each other and each sentence has 3 true positives and 3 false positives. With existing systems, both queries would provide the same hits as they treat sentences as conjunctions of words.

2. A video search engine that supports all 10 full-length Hollywood movies and all 141 sentential queries discussed in the text is available at http://0xab.com/research/video-events/westerns.html.

The backpack approached the trash can.
The red object approached the chair.
The person to the left of the trash can put down an object.
The person put down the trash can.
The person carried the red object.
The person picked up an object to the left of the trash can.
The person picked up an object.
The person picked up an object quickly.
The person carried an object towards the trash can.
The backpack approached the chair.
The person put down the chair.

The chair approached the trash can.
The blue object approached the chair.
The person to the right of the trash can put down an object.
The person put down the backpack.
The person carried the blue object.
The person picked up an object to the right of the trash can.
The person put down an object.
The person picked up an object slowly.
The person carried an object away from the trash can.
The red object approached the trash can.

Fig. 6. The 21 sentential queries used in Section 6.1. Differences in corresponding minimal pairs are highlighted in red and green.

The person carried an object away from the trash can.

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

Fig. 7. Sentential-query-based video search: returning the best-scoring video, in a corpus of 94 videos, for a given sentence.

```
X {approached Y {,quickly,slowly} {,from the left,from the right},
   {lead,rode} Y {,quickly,slowly} {,leftward,rightward, {towards,away from} Z}}
```

Fig. 8. The template used to generate the 141 query sentences where X, Y, and Z are either *person* or *horse*. The template generates 204 sentences out of which 63 are removed because they involve people riding people and horses riding people or other horses for which no true positives exist in our video corpus.

6.3 Comparison

We compare our results against a baseline method that employs the same approach that is used in state-of-the-art video-search systems. We do not compare against any particular existing system because no current system employs state-of-the-art object or event detectors and thus any such system would be severely handicapped in its inability to reliably detect people, horses, and the particular events we search for. Our baseline operates as follows. We first apply an object detector to each frame of every clip to detect people and horses. For comparison purposes, we employ the same object detector and pretrained models as used for the experiments in Section 6.2, including passing the raw detector score through the same sigmoid. We rank the clips by the average score of the top detection in each frame. If the query sentence contains only the word *person*, we rank only by the person detections. If the query sentence contains only the word *horse*, we rank only by the horse detections. If the query sentence contains both the words *person* and *horse*, we rank by the average of the top person and top horse detection in each frame. We then apply a binary event detector to eliminate clips from the ranking that do not depict the event specified by the entire query

sentence. For this purpose, we employ a state-of-the-art event detector, namely that of Kuehne *et al.* [16]. We train that detector on six samples of each entire query sentence and remove those samples from the test set. We then report the top 10 ranked clips that satisfy the event detector and compare those clips against the top 10 clips produced by our method.

We compared our system against this baseline on three different sentential queries: *The person rode the horse*, *The person lead the horse*, and *The person approached the horse*. The results are summarized in Fig. 9. Note that our approach yields significantly higher precision on each of the queries as well as higher overall average precision. Further note that this baseline system was trained on a total of 18 training samples: six samples for each of three query sentences In contrast, our method was trained on a total six training samples. Not only was our method trained on one third as many training samples, our method can support all 141 distinct queries with its training set, while the baseline only supports three queries with its training set.

7 Discussion

As discussed in Section 1, previous work falls into two categories: search by example and attribute-based approaches. In the former, a sample image or video is provided and similar images or videos are retrieved. Conventional event-recognition systems are of this type. They train models on collections of query clips and find the target clips which best match the trained model. In the limit, such systems find the target clips most-similar to a single query clip. Attribute-based approaches are usually applied to images, not videos. Such approaches, given a sentence or sentence fragment, extract the words from that sentence and use independent word models to score each image or video clip [33], [34]. Some variants of these approaches, such as that of Siddiquie *et al.* [35], learn correlations between multiple features and include feature detectors which are not present in the input query. Some systems present various combinations of the approaches described above such as those of Christel *et al.* [17], Worring *et al.* [18], and Snoek *et al.* [19].

None of the approaches described above link features in a way that is informed by the structure of the sentence, hence they are unable to support sentential queries. What we mean by this is they cannot show the key difference that we underscore in this work, the ability to encode the semantics of a query sentence with enough fidelity to differentiate between *The person rode the horse* and *The horse rode the person*. The baseline system we compare against in Section 6.3 was specifically designed to model the predominant current methodology, updated to use state-of-the-art object and event recognizers. Specifically, it modeled queries as bags of words with no reflection of argument structure.

In the experiments in Section 6.2, we report only true positives and the associated precision, not true negatives nor the associated recall. The reason is simple: reporting true negatives would require annotating the entire corpus of 37187 clips with truth values for all 141 queries, a monumental and tedious task. We only annotate the top ten hits for each of the 141 queries as to their truth value, allowing us only to report true positives. That raises a potential question: what is the chance that we may have missed potential hits for our queries. We note that movies have very different properties from surveillance video and standard action-recognition corpora. Most time is spent showing people engaged in dialog rather than performing actions. Thus we contend that the false negative rate is very low. Moreover, we contend that chance performance on this retrieval task is also very low. This is further supported by the extreme low performance of the baseline from Section 6.3. Thus we contend that the underlying retrieval task is difficult and the performance of our method as described in Section 6.2 is good. Moreover, we have annotated negatives for the smaller experiment in Section 6.1 which allowed us to compute chance performance and demonstrate that our method far exceeds such.

In the future, one can imagine scaling our approach along a variety of axes: larger and more varied video corpora, a larger lexicon of nouns, verbs, adjectives, adverbs, and prepositions, and a more complex query grammar. Let us consider the advances needed to achieve such scaling.

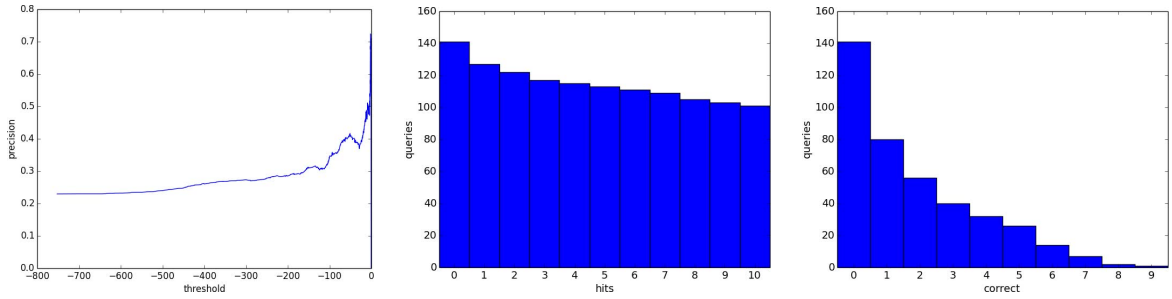
Scaling the size of the video corpus is easy. For a fixed-size query language, processing time and space is linear in the corpus size. Further, such processing is trivially parallelizable and, as discussed in Section 5, many components of the process, such as object detection, can be precomputed and cached in a query-independent fashion. Moreover, as discussed in Section 5, results of earlier queries can be cached and used to speed up processing of later queries, potentially leading to reduction of the search complexity below linear time.

Scaling up to support a larger lexicon of nouns, largely depends on the state-of-the-art in object detection. While current methods appear to work well only for small numbers of object classes, recent work by Dean *et al.* [36] has shown that object detection may scale to far larger collections of objects. Since our method simply requires scored detections, it can avail itself of any potential future advances in object detection, including combining the results of multiple detection methods, potentially even for the same object class as part of the same object track.

Scaling up to support a larger lexicon of verbs also appears possible. Our approach performs event recognition on time series of feature vectors extracted from object tracks. This general approach has already been demonstrated to scale to 48 distinct event classes [29]. However, this can only be used for verbs and other parts of speech whose meanings are reflected in motion profile: the changing relative and absolute positions, velocities, and accelerations of the event participants. Scaling beyond this, to encode the meanings of words like *sit*, *pour*, *build*, or *break*, or semantic distinctions like

Query	our TP	baseline TP
The person rode the horse.	9	0
The person lead the horse.	1	0
The person approached the horse.	4	1

Fig. 9. A comparison between our approach and a baseline system constructed out of state-of-the-art components on the top 10 hits returned for various sentential queries.



(a) (b) (c)

Fig. 10. (a) Average precision of the top 10 hits for the 141 query sentences as a function of the threshold on the sentence-tracker score. Without a threshold, (b) the number of sentences with at most the given number of hits and (c) the number of sentences with at least the given number of correct hits.

the difference between *abandon* and *leave* or between *follow* and *chase*, would require modeling facets of human perception and cognition beyond motion profile, such as body posture [37], functionality, intention, and physical processes.

Scaling sentence length and complexity requires lattices of greater width. The dynamic-programming algorithm which performs inference on the sentence-tracker lattice takes time quadratic in the width of the cross-product lattice. Unfortunately the width of this cross-product lattice increases exponentially in the number of participants and the query-sentence length. While this approach will not scale indefinitely, it is able to process our current queries with three participants and 6-14 words with an acceptable runtime, a few dozen seconds per clip. Scaling further will require either a faster dynamic-programming algorithm or inexact inference. Barbu *et al.* [28] present an algorithm which employs Felzenszwalb and Huttenlocher's [38] generalized distance transform to perform inference in linear time in the lattice width, as opposed to quadratic time, for a one-word sentence tracker. Such an approach can be generalized to an entire sentence tracker but carries the added weight of restricting the form of the features that are extracted from tracks when formulating the per-state predicates in the event model. At present, the constant-factor overhead of this approach outweighs the reduced asymptotic complexity, but this may change with increased query-sentence complexity. Alternatively one might perform inexact inference using beam search to eliminate low-scoring lattice regions. Inexact inference might also employ sampling methods such as MCMC. Lazy Viterbi [39] offers another alternative which maintains the optimality of the algorithm but only visits nodes in the lattice as needed.

8 Conclusion

We have developed an approach to video search which takes as input a video corpus and a sentential query. It generates a list of results ranked by how well they depict the query sentence. This approach provides two novel video-search capabilities. First, it can encode the semantics of sentences compositionally, allowing it to express subtle distinctions such as the difference between *The person rode the horse* and *The horse rode the person*. Such encoding allows it to find depictions of novel sentences which have never been seen before. Second, it extends video search past nouns and verbs allowing sentences which can encode modifiers such as adverbs and entire prepositional phrases. Unlike other approaches which allow for textual queries of images or videos, we do not require any prior video annotation. The entire lexicon shares a small number of parameters and, unlike previous work, this approach does not need to be trained on every word or even every related word. We have evaluated this approach on a large video corpus of 10 Hollywood movies, comprising roughly 20 hours of video, by running 141 sentential queries.

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.

(a) *The horse approached the person.*

(b) *The person approached the horse.*

Fig. 11. Frames from the top 6 hits for two sentential queries. True positives are shown in green and false positives in red. In both cases, half are true positives[1]. The fact that the results are different shows that our method encodes the meaning of the entire sentence along with which object fills which role in that sentence.

REFERENCES

[1] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A Survey on Visual Content-Based Video Indexing and Retrieval," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, no. 6, pp. 797–819, 2011.

[2] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Baby Talk: Understanding and Generating Simple Image Descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1601–1608.

[3] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[4] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan, "Trajectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video," in *Proceedings of the ACM International Conference on Multimedia*, 2003, pp. 11–20.

[5] A. Anjulan and N. Canagarajah, "A Unified Framework for Object Retrieval and Mining," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 1, pp. 63–76, 2009.

[6] D. Byrne, A. R. Doherty, C. G. M. Snoek, G. J. F. Jones, and A. F. Smeaton, "Everyday Concept Detection in Visual Lifelogs: Validation, Relationships, and Trends," *Multimedia Tools and Applications*, vol. 49, no. 1, pp. 119–144, 2010.

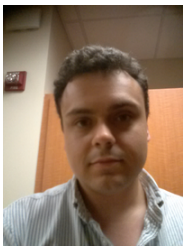
[7] M. A. Sadeghi and A. Farhadi, "Recognition using Visual Phrases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1745–1752.

[8] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III, "Midge: Generating Image Descriptions from Computer Vision Detections," in *Proceedings of the Conference of European Chapter of the Association for Computational Linguistics*, 2012, pp. 747–756.

[9] C. G. Snoek and M. Worring, "Concept-Based Video Retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2008.

[10] P. Chang, M. Han, and Y. Gong, "Extract Highlights from Baseball

Game Video with Hidden Markov Models," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, 2002, pp. 609–612.
[11] S. Sadanand and J. J. Corso, "Action Bank: A High-Level Representation of Activity in Video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1234–1241.
[12] C.-Y. Chen and K. Grauman, "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*, 2013, pp. 572–579.
[13] Y. Song, L.-P. Morency, and R. Davis, "Action Recognition by Hierarchical Sequence Summarization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3562–3569.
[14] I. Everts, J. C. van Gemert, and T. Gevers, "Evaluation of Color STIPs for Human Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2850–2857.
[15] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal Deformable Part Models for Action Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2642–2649.
[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2556–2563.
[17] M. G. Christel, C. Huang, N. Moraveji, and N. Papernick, "Exploiting Multiple Modalities for Interactive Video Retrieval," in *Proceedings of the International Conference on Accoustics, Speech, and Signal Processing*, vol. 3, 2004, pp. 1032–1035.
[18] M. Worring, C. G. Snoek, O. De Rooij, G. Nguyen, and A. Smeulders, "The MediaMill Semantic Video Search Engine," in *Proceedings of the International Conference on Accoustics, Speech, and Signal Processing*, vol. 4, 2007, pp. 1213–1216.
[19] C. G. Snoek, M. Worring, D. C. Koelma, and A. W. Smeulders, "A Learned Lexicon-Driven Paradigm for Interactive Video Retrieval," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 280–292, 2007.
[20] Y. Aytar, M. Shah, and J. Luo, "Utilizing Semantic Word Similarity Measures for Video Retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
[21] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade Object Detection with Deformable Part Models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2241–2248.
[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
[23] H. O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, and T. Darrell, "Sparselet Models for Efficient Multiclass Object Detection," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 802–815.
[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[25] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
[26] A. J. Viterbi, "Error bounds for convolutional codes and an asymtotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–267, Apr. 1967.
[27] ——, "Convolutional Codes and their Performance in Communication Systems," *IEEE Transactions on Communication*, vol. 19, no. 5, pp. 751–772, 1971.
[28] A. Barbu, N. Siddharth, A. Michaux, and J. M. Siskind, "Simultaneous Object Detection, Tracking, and Event Recognition," *Advances in Cognitive Systems*, vol. 2, pp. 203–220, 2012.
[29] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. J. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. M. Siskind, J. W. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang, "Video In Sentences Out," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2012, pp. 102–112.
[30] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 423–430.
[31] M. Cooper, T. Liu, and E. Rieffel, "Video Segmentation via Temporal Pattern Classification," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 610–618, 2007.
[32] H. Yu and J. M. Siskind, "Grounded Language Learning from Video Described with Sentences," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 53–56.
[33] D. A. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk, "Attribute-Based People Search in Surveillance Environments," in *Workshop on Applications of Computer Vision*, 2009, pp. 1–8.
[34] N. Kumar, P. Belhumeur, and S. Nayar, "FaceTracer: A Search Engine for Large Collections of Images with Faces," in *Proceedings of the European Conference on Computer Vision*, 2008, pp. 340–353.
[35] B. Siddiquie, R. S. Feris, and L. S. Davis, "Image Ranking and Retrieval Based on Multi-Attribute Queries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 801–808.
[36] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, Accurate Detection of 100,000 Object Classes on a Single Machine," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1814–1821.
[37] L. Bourdev and J. Malik, "Poselets: Body Part Detectors Trained using 3D Human Pose Annotations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1365–1372.
[38] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance Transforms of Sampled Functions," Cornell Computing and Information Science, Tech. Rep. TR2004-1963, 2004.
[39] J. Feldman, I. Abou-Faycal, and M. Frigo, "A Fast Maximum-Likelihood Decoder for Convolutional Codes," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, 2002, pp. 371–375.



Andrei Barbu received the BCS degree from the University of Waterloo in 2008. He is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University. His research interests lie at the intersection of computer vision, natural language, and robotics. He is particularly interested in how both machines and humans can use language to transfer knowledge between multiple modalities and reason across both language and vision simultaneously.

N. Siddharth received the B.E. degree in electronics and communication engineering from Anna University, Chennai, India in 2008 and is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University. His research interests include artificial intelligence, computer vision, computational linguistics, machine learning, robotics, cognitive science, and cognitive neuroscience.

Jeffrey Mark Siskind received the B.A. degree in computer science from the Technion, Israel Institute of Technology in 1979, the S.M. degree in computer science from MIT in 1989, and the Ph.D. degree in computer science from MIT in 1992. He did a postdoctoral fellowship at the University of Pennsylvania Institute for Research in Cognitive Science from 1992 to 1993. He was an assistant professor at the University of Toronto Department of Computer Science from 1993 to 1995, a senior lecturer at the Technion Department of Electrical Engineering in 1996, a visiting assistant professor at the University of Vermont Department of Computer Science and Electrical Engineering from 1996 to 1997, and a research scientist at NEC Research Institute, Inc. from 1997 to 2001. He joined the Purdue University School of Electrical and Computer Engineering in 2002 where he is currently an associate professor. His research interests include machine vision, artificial intelligence, cognitive science, computational linguistics, child language acquisition, and programming languages and compilers.