Video Retrieval with Sentential Queries

Andrei Barbu* andrei@0xab.com
N. Siddharth* siddharth@iffsid.com
Jeffrey Mark Siskind* qobi@purdue.edu

Abstract

We present an approach to searching large video corpora for video clips which depict a natural-language query in the form of a sentence. This approach uses compositional semantics to encode subtle meaning that is lost in other systems, 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 video-sentence pair and a natural-language parser, along with a grammar that describes the space of sentential queries, we produce a score which indicates how well the video depicts the sentence. We produce such a score for each video clip in a corpus and return a ranked list of clips. Furthermore, this approach addresses two fundamental problems simultaneously: detecting and tracking objects, and recognizing whether those tracks depict the query. Because both tracking and object detection are unreliable, this uses knowledge about the intended 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 show how one can 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 queries involving people and horses interacting with each other in 10 full-length Hollywood movies.*

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 these problems. Systems 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 the 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 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.

Hu et al. [12] note that recent work on semantic video search focuses on detecting nouns, detecting verbs, or using language to search already-existing video annotation. Work that detects objects does not employ object detectors, but instead relies on statistical features to cluster videos with similar objects. Sivic and Zisserman [14] extract local features from a positive example of an object to find key frames that contain the same object. Anjulan and Canagarajah [1] 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

*School of Electrical and Computer Engineering, Purdue University, West Lafayette IN 47907-2035

search a collection of videos for a particular object class but instead must search by example.

Prior work on verb detection does not integrate with work on object detection. Chang et al. [5] find one of four different highlights in basketball games using hidden Markov models and the expected structure of a basketball game. It does not detect objects but instead classifies entire presegmented clips, is restricted to a small number of domain-specific actions, and supports only single-word queries. Yu et al. [21] track one object, a soccer ball, and detect actions being performed on that object during a match by the position and velocity of the object. It supports a small number of domain-specific actions and is limited to a single object. In summary, the above approaches only allow for searching for a single word, a verb, and are domain-specific.

Prior work on more complex queries involving both nouns and verbs essentially encodes the meaning of a sentence as a conjunction of words, discarding the semantics of the sentence. Christel et al. [6], Worring et al. [18], and Snoek et al. [15] 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. [2] 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, it retrieves a ranked list of videos which are described by that sentence. We show a method of 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 novel way to combine the semantics of words into the semantics of sentences and to combine sentence recognition with object tracking in order to score a video-sentence pair. To demonstrate this approach, we run 141 natural-language queries of a corpus of 10 full-length Hollywood movies using a grammar which includes nouns, verbs, adverbs, and spatial-relation 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

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. [10, 11] object detector, specifically the variant developed by Song et al. [16]. 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% [9]. 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 and 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. Each detection j has an associated axis-aligned rectangle b_j^t and score $f(b_j^t)$ and each pair of detections has an associated temporal coherence score $g(b_{j^{t-1}}^{t-1}, b_{j^t}^t)$ where t is the index of the current frame in a video of length T. We formulate the score of a track $\mathbf{j} = \langle j^1, \ldots, j^T \rangle$ as

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

where we take g, the motion coherence, to be a 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, the Viterbi [17] algorithm.

The development of this tracker follows that of Barbu et al. [3] which presents additional details of such a tracker, including an extension which allows it to generate multiple tracks per object class by non-maxima suppression. The tracker employed here has a number of differences from that

of Barbu et al. [3]. While that tracker used the raw detection scores from the Felzenszwalb et al. [10, 11] detector, these scores are difficult to interpret because the mean score and variance 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 all FSMs used in this paper, rendered as regular expressions along with their semantics. This approach is a limiting case of that taken by Barbu et al. [4] which used hidden Markov models (HMMs) to encode the semantics of words. 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. [4] to be employed here.

We construct word meaning 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. The reason for this two-level construction is to allow for sharing of low-level features and parameters. All words share the same predicates which are encoded relative to 9 parameters: **far**, **close**, **stationary**, Δ**closing**, Δ**angle**, Δ**pp**, Δ**quickly**, Δ**slowly**, and **overlap**. These parameters are learned from a small number of positive and negative examples that cover only a small number of 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 vary with the resolution.

Given a regular expression for a word, we can construct a non-deterministic 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

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

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 will be zero. If it does not, the score will be $-\infty$. The above formulation is trivially generalized to multiple tracks and is the same as that used by Barbu et al. [3]. We find accepting paths through the lattice of states 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 *left-of*. We will avail ourselves of the ability to encode the meaning of all parts of speech into a uniform representation in order to build up the semantics of sentences from the semantics of words.

4. Sentence tracker

Our ultimate goal is to search for videos given 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, we want a person that is both *running* and doing so *leftward*. 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 semantics of a sentence. Intuitively, we need a way to 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 detections and states, respectively, and each weights the links in that lattice by a score, the motion coherence and state-transition score, respectively. We simultaneously find the optimal, highest-scoring, track $j^1, \ldots, j^T$ and state

$$
\begin{array}{lcl}
\textsc{far}(a,b) & \triangleq & |a_{cx} - b_{cx}| - \frac{a_{width}}{2} - \frac{b_{width}}{2} > \textbf{far} \\
\textsc{really-close}(a,b) & \triangleq & |a_{cx} - b_{cx}| - \frac{a_{width}}{2} - \frac{b_{width}}{2} > \frac{\textbf{close}}{2} \\
\textsc{close}(a,b) & \triangleq & |a_{cx} - b_{cx}| - \frac{a_{width}}{2} - \frac{b_{width}}{2} > \frac{\textbf{close}}{2} \\
\textsc{stationary}(b) & \triangleq & \text{flow-magnitude}(b) \leq \textbf{stationary} \\
\textsc{closing}(a,b) & \triangleq & |a_{cx} - b_{cx}| > |\text{project}(a)_{cx} - \text{project}(b)_{cx}| + \Delta\textbf{closing} \\
\textsc{departing}(a,b) & \triangleq & |a_{cx} - b_{cx}| < |\text{project}(a)_{cx} - \text{project}(b)_{cx}| + \Delta\textbf{closing} \\
\textsc{moving-direction}(a,b,\alpha) & \triangleq & |\text{flow-orientation}(a) - \alpha|^\circ < \Delta\textbf{angle} \wedge \\
 & & \text{flow-magnitude}(a) > \textbf{stationary} \\
\textsc{left-of}(a,b) & \triangleq & a_{cx} < b_{cx} + \Delta\textbf{pp} \\
\textsc{right-of}(a,b) & \triangleq & a_{cx} > b_{cx} + \Delta\textbf{pp} \\
\textsc{leftward}(a,b) & \triangleq & \textsc{moving-direction}(a,b,0) \\
\textsc{leftward}(a,b) & \triangleq & \textsc{moving-direction}(a,b,\pi) \\
\textsc{stationary-but-far}(a,b) & \triangleq & \textsc{far}(a,b) \wedge \textsc{stationary}(a) \wedge \textsc{stationary}(b) \\
\textsc{stationary-but-close}(a,b) & \triangleq & \textsc{close}(a,b) \wedge \textsc{stationary}(a) \wedge \textsc{stationary}(b) \\
\textsc{moving-together}(a,b) & \triangleq & |\text{flow-orientation}(a) - \text{flow-orientation}(b)|^\circ < \Delta\textbf{angle} \wedge \\
 & & \text{flow-magnitude}(a) > \textbf{stationary} \wedge \\
 & & \text{flow-magnitude}(b) > \textbf{stationary} \\
\textsc{approaching}(a,b) & \triangleq & \textsc{closing}(a,b) \wedge \textsc{stationary}(b) \\
\textsc{quickly}(a) & \triangleq & \text{flow-magnitude}(a) > \Delta\textbf{quickly} \\
\textsc{slowly}(a) & \triangleq & \textbf{stationary} < \text{flow-magnitude}(a) < \Delta\textbf{slowly} \\
\textsc{overlapping}(a,b) & \triangleq & \frac{a \cap b}{a \cup b} \geq \textbf{overlap}
\end{array}
$$

Figure 1. Predicates which accept detections, denoted by a and b, formulated around 9 parameters. 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.

$$
\begin{array}{lcl}
\texttt{horse}(a) & \triangleq & (a_{\text{object-class}} = \text{“}horse\text{''})^+ \\
\texttt{person}(a) & \triangleq & (a_{\text{object-class}} = \text{“}person\text{''})^+ \\
\texttt{quickly}(a) & \triangleq & \textbf{true}^+ \; \textsc{quickly}(a)^{\{3,\}} \; \textbf{true}^+ \\
\texttt{slowly}(a) & \triangleq & \textbf{true}^+ \; \textsc{slowly}(a)^{\{3,\}} \; \textbf{true}^+ \\
\texttt{from the left}(a,b) & \triangleq & \textbf{true}^+ \; \textsc{left-of}(a,b)^{\{5,\}} \; \textbf{true}^+ \\
\texttt{from the right}(a,b) & \triangleq & \textbf{true}^+ \; \textsc{right-of}(a,b)^{\{5,\}} \; \textbf{true}^+ \\
\texttt{leftward}(a) & \triangleq & \textbf{true}^+ \; \textsc{leftward}(a)^{\{5,\}} \; \textbf{true}^+ \\
\texttt{rightward}(a) & \triangleq & \textbf{true}^+ \; \textsc{rightward}(a)^{\{5,\}} \; \textbf{true}^+ \\
\texttt{to the left of}(a,b) & \triangleq & \textbf{true}^+ \; \textsc{left-of}(a,b)^{\{3,\}} \; \textbf{true}^+ \\
\texttt{to the right of}(a,b) & \triangleq & \textbf{true}^+ \; \textsc{right-of}(a,b)^{\{3,\}} \; \textbf{true}^+ \\
\texttt{towards}(a,b) & \triangleq & \textsc{stationary-but-far}(a,b)^+ \; \textsc{approaching}(a,b)^{\{3,\}} \\
 & & \textsc{stationary-but-close}(a,b)^+ \\
\texttt{away from}(a,b) & \triangleq & \textsc{stationary-but-close}(a,b)^+ \; \textsc{departing}(a,b)^{\{3,\}} \\
 & & \textsc{stationary-but-far}(a,b)^+ \\
\texttt{ride}(a,b) & \triangleq & \textbf{true}^+ \; (\textsc{moving-together}(a,b) \wedge \textsc{overlapping}(a,b))^{\{5,\}} \; \textbf{true}^+ \\
\texttt{lead}(a,b) & \triangleq & \textbf{true}^+ \left(\begin{array}{l} \neg\textsc{really-close}(a,b) \wedge \\ \textsc{moving-together}(a,b) \wedge \\ \left((\textsc{left-of}(a,b) \wedge \textsc{leftward}(a)) \vee \right. \\ \left. (\textsc{right-of}(a,b) \wedge \textsc{rightward}(a)) \right) \end{array} \right)^{\{5,\}} \textbf{true}^+ \\
\texttt{approach}(a,b) & \triangleq & \textbf{true}^+ \; \textsc{approaching}(a,b)^{\{5,\}} \; \textbf{true}^+
\end{array}
$$

Figure 2. Regular expressions which encode the meanings of each of the 15 words or lexicalized phrases in the lexicon. These are composed from the predicates shown in Fig. 1. We use an extended regular-expression syntax where an exponent of $\{t,\}$ allows a predicate to hold for t or more frames.

sequence $k^1, \ldots, k^T$ as

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

which ensures that, unless the state sequence for the word FSM leads to an accepting state, the resulting score will be $-\infty$ and thereby constrains the tracks to depict the word. 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 a cross-product lattice.

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. 4, and extract the number of participants and the track-to-role mapping. Each

sentence has a number of thematic roles that must be filled by participants in order for the sentence to be syntactically valid. 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*, *patient*, and *goal*. Each word in this sentence refers to a subset of these three different participants, as shown in Fig. 3(right), and words that refer to multiple participants, such as *ride*, must be assigned participants in the correct 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_w^n, map the nth role in a word w to a tracker. Fig. 3(right) 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_w^n, we find a collection of optimal tracks $\langle j_1^1, \ldots, j_1^T \rangle \cdots \langle j_L^1, \ldots, j_L^T \rangle$, one for each participant, and accepting state sequences $\langle k_1^1, \ldots, k_1^T \rangle \cdots \langle k_W^1, \ldots, k_W^T \rangle$, one for each word, as

$$\max_{\substack{j_1^1,\ldots,j_1^T \\ \vdots \\ j_L^1,\ldots,j_L^T}} \max_{\substack{k_1^1,\ldots,k_1^T \\ \vdots \\ k_W^1,\ldots,k_W^T}} \left(\sum_{l=1}^{L} \sum_{t=1}^{T} f(b_{j_l^t}^t) + \sum_{t=2}^{T} g(b_{j_l^{t-1}}^{t-1}, b_{j_l^t}^t) + \sum_{w=1}^{W} \sum_{t=1}^{T} h_w(k_w^t, b_{j_{\theta_w^1}^t}^t, b_{j_{\theta_w^2}^t}^t) + \sum_{t=2}^{T} a_w(k_w^{t-1}, k_w^t) \right) \quad (3)$$

where a_w and h_w are the transition matrices and predicates for word w, $b_{j_l^t}^t$ is a detection in the tth frame of the lth track, and $b_{j_{\theta_w^n}^t}^t$ connects a participant that fills the nth role in word w with the detections of its tracker. This equation maximizes the tracker score for each tracker corresponding to each participant, and ensures that each word has a sequence of accepting states, if such a sequence exists, otherwise the entire sentence-tracker score will be $-\infty$. In essence, we are taking cross products of tracker lattices and word lattices while ensuring that the sequence of cross products agrees with our track-to-role mapping and finding 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 means paying a penalty by having to choose lower-scoring detections.

5. Results

We have so far developed a system which scores a video-sentence pair telling us how well a video depicts a sentence. Given a sentential query, we run the sentence tracker on every video in a corpus and return all results ranked by their scores. The better the score the more confident we are that the resulting tracks correspond to real objects in the video while the sentence tracker itself ensures that all tracks produced satisfy the sentential query. To save on redundant computation, we cache the object-detector results for each video as the detection scores are independent of the sentential query.

To demonstrate this approach to video search, we ran sentential queries over a corpus 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 downsample all videos to 6 frames per second but keep 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, which overlaps 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 [7] we do 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 up the videos into clips is to return multiple such events.

We adopt the grammar from Fig. 4 which 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 are not testing on the training data, we employ previously-trained object models that have not been trained on these videos but have instead been trained on the PASCAL VOC Challenge [9]. 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 train all 9 parameters simultaneously on only 3 positive examples and 3 negative examples. Note

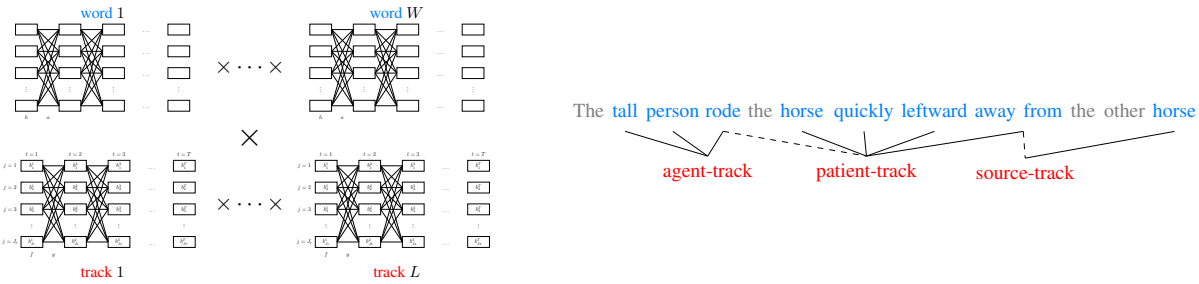


Figure 3. (left) Tracker lattices are used to produce tracks for each the object. Word lattices constructed from word FSMs recognize one or more tracks. We take the cross product of multiple tracker lattices and word lattices to simultaneously track objects and recognize words. By construction, this ensures that the resulting tracks are described by the desired words. (right) 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.

S	$\rightarrow$	NP VP	NP	$\rightarrow$	D N [PP]
D	$\rightarrow$	*the*	N	$\rightarrow$	*person* | *horse*
PP	$\rightarrow$	P NP	P	$\rightarrow$	*to the left of* | *to the right of*
VP	$\rightarrow$	V NP [Adv] [PP_M]	V	$\rightarrow$	*lead* | *rode* | *approached*
Adv	$\rightarrow$	*quickly* | *slowly*	PP_M	$\rightarrow$	P_M NP | *from the left* | *from the right*
P_M	$\rightarrow$	*towards* | *away from*			

Figure 4. The grammar for sentential queries used in the experiment section.

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 proceeds 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 classifies all 6 training samples which are then removed from the test set. Yu and Siskind [20] present a related alternative 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. 4 from the template shown in Fig. 5. We eliminate 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 score every video-sentence pair 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. 7(a), we show the precision of the system on the top 10 queries 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. 7(b), we show the number of results returned per sentence, eliminating those results which have a score of $-\infty$ since that tells us 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. 7(c), we show the number of correct his 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. 8 where we show the top 6 hits for two similar queries: *The person approaches the horse* and *The horse approached the person*. Hits are presented in order of score, with the highest scoring hit in the top left-hand corner and scores decreasing as one moves to the right and to the next line. 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.[1] With existing systems, both queries would provide the same hits as they treat the sentences as conjunctions of words.

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

[1] It can be difficult to distinguish true positives and false positives from just a pair of frames, so we have included the full videos in the supplementary material along with additional results.

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

Figure 5. 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.

object detector and pretrained models as used for the earlier experiment, 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*. [13]. 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. 6. 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.

6. Conclusion

We have developed a framework for a novel kind of video search that takes, as input, natural-language queries in the form of sentences, along with a video corpus, and generates a list of ranked results. 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*. Second, it can also encode structures more complex than just nouns and verbs, such as modifiers, *e.g*. adverbs, and entire phrases, *e.g*. prepositional phrases. 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 and annotating the top 10 results for each query.

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.

References

[1] 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.

[2] Y. Aytar, M. Shah, and J. Luo, "Utilizing semantic word similarity measures for video retrieval," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[3] 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.

[4] 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 Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012, pp. 102–112.

[5] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden Markov models," in *Proceedings of the International Conference on Image Processing*, vol. 1, 2002, pp. 609–612.

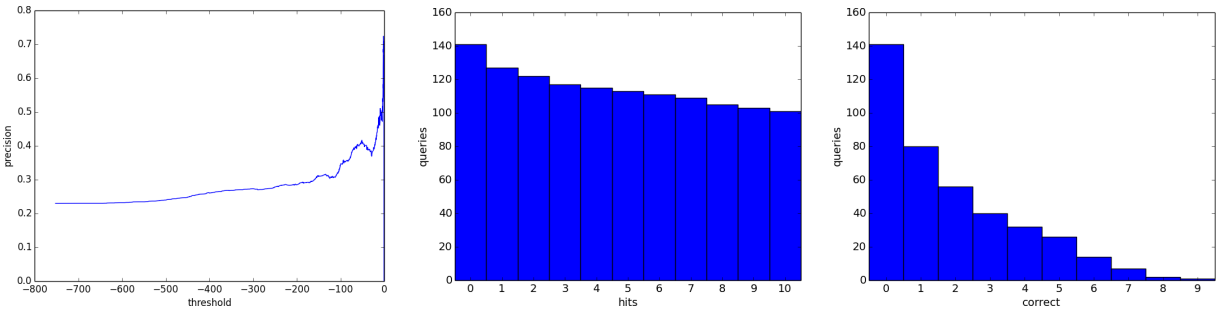
[6] M. G. Christel, C. Huang, N. Moraveji, and N. Papernick, "Exploiting multiple modalities for interactive video retrieval," in *Proceedings of the International Conference on Accoustics, Speech, and Signal Processing*, vol. 3, 2004, pp. 1032–1035.

[7] 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.

[8] L.-Y. Duan, M. Xu, Q. Tian, C.-S. Xu, and J. S. Jin, "A unified framework for semantic shot classification in sports video," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1066–1083, 2005.

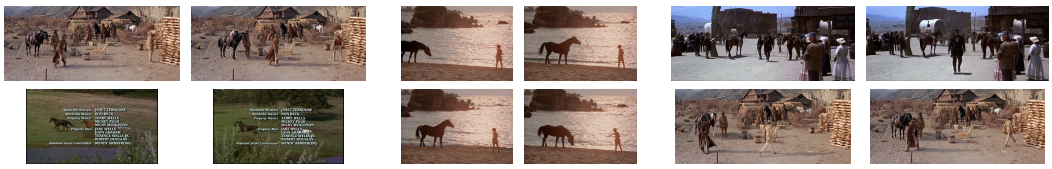
query	our TP	baseline TP
The person rode the horse.	9	0
The person lead the horse.	1	0
The person approached the horse.	4	1

Figure 6. 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)

Figure 7. (a) Average precision of the top 10 hits for the 141 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.



(a) *The horse approached the person*

(b) *The person approached the horse*

Figure 8. The top 6 hits for two sentences. 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.

[9] 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.

[10] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2241–2248.

[11] 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, Sep. 2010.

[12] 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, Part C: Applications and Reviews*, vol. 41, no. 6, pp. 797–819, 2011.

[13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision*, 2011.

[14] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[15] 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.

[16] 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*. Springer, 2012, pp. 802–815.

[17] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transactions on Communication*, vol. 19, pp. 751–772, Oct. 1971.

[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] G. Xu, Y.-F. Ma, H.-J. Zhang, and S.-Q. Yang, "An HMM-based framework for video semantic analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1422–1433, 2005.

[20] H. Yu and J. M. Siskind, "Grounded language learning from video described with sentences," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[21] 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 Eleventh ACM International Conference on Multimedia*, 2003, pp. 11–20.