A Markov Logic Framework for Recognizing Complex Events from Multimodal Data

Young Chol Song, Henry Kautz, James Allen, Mary Swift, Yuncheng Li, Jiebo Luo
Department of Computer Science, University of Rochester
Rochester, NY, USA
{ysong, kautz, james, swift, yli, jluo}@cs.rochester.edu

ABSTRACT

We present a general framework for complex event recognition that is well-suited for integrating information that varies widely in detail and granularity. Consider the scenario of an agent in an instrumented space performing a complex task while describing what he is doing in a natural manner. The system takes in a variety of information, including objects and gestures recognized by RGB-D and descriptions of events extracted from recognized and parsed speech. The system outputs a complete reconstruction of the agent's plan, explaining actions in terms of more complex activities and filling in unobserved but necessary events. We show how to use Markov Logic (a probabilistic extension of first-order logic) to create a model in which observations can be partial, noisy, and refer to future or temporally ambiguous events; complex events are composed from simpler events in a manner that exposes their structure for inference and learning; and uncertainty is handled in a sound probabilistic manner. We demonstrate the effectiveness of the approach for tracking kitchen activities in the presence of noisy and incomplete observations.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: 3D/stereo scene analysis; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Predicate logic; I.2.3 [**Deduction and Theorem Proving**]: Probabilistic reasoning; I.2.7 [**Natural Language Processing**]: Language parsing and understanding

General Terms

Human Factors

Keywords

Multimodal Interaction; Plan Recognition; Markov logic

ICMI'13, December 9–13, 2013, Sydney, Australia

ACM 978-1-4503-2129-7/13/12 ...$15.00.
http://dx.doi.org/10.1145/2522848.2522883.

1. INTRODUCTION

Consider a situation where you are observing a person demonstrating both physically and verbally how to perform a complex task, for example, preparing a cup of tea. The subject performs simple actions (*e.g.,* picking up a tea kettle), which are part of more complex activities (filling the kettle from the sink), and which in turn are part of yet higher-level activities (preparing hot water), *ad infinitum.* Actions are inextricably connected to changes in the state of the world (moving the cup changes its location), even if the change is not directly observable (stirring the tea after adding sugar dissolves the sugar). The subject may refer to actions in the past ("I've finished filling the kettle..."), the current moment ("The water is heating..."), or in the future ("I still have to get the milk and sugar..."), and complex events can overlap temporally (the subject fetches the tea box while the water is heating). The subject may describe an event at different levels of abstraction ("I'll heat water" *vs* "I'll heat water in the microwave"), or provide a partial verbal description, which is resolved by context ("Now I pour the water [from the kettle into the cup]"). Similarly, visual percepts of events may be incomplete due to visual resolution or obscuring objects, and only disambiguated by context (hand removes *something* from tea box).

A human observer reflexively tries to understand the situation by explaining what he sees and hears in terms of the subject's *plan*: a coherent, connected structure of observed, hypothesized, and predicted structure of actions and properties. When the subject is a teacher, the latter must piece together a new plan. In other cases, the plan is one familiar to the observer, whose task becomes identifying, instantiating, and tracking the plan; such is the case, *e.g.,* when a teacher observes a student at work. For this thought exercise, we focused on cooking, but the same principles apply to many domains where there is a practical need for automated plan recognition, such as wet labs, medical procedures, equipment maintenance, and surveillance.

While there is a rich history of research on plan recognition (briefly recapped in the next section), most work makes assumptions about the nature of actions and observations that are violated by the simple example above. We argue that a general framework for plan recognition should meet the following criteria: (i) Be robust across variations in the appearance of a scene and the language used to describe it: *i.e.,* provide a semantic as opposed to an appearance model. (ii) Support easy knowledge engineering, *e.g.,* for defining events in terms of changes of properties of objects and/or collections of other events. (iii) Represent both decomposi-

tion and abstraction event hierarchies, with no fixed number of levels. (iv) Treat instances of events as entities to which reference can be made: *e.g.,* support event reification. (v) Allow events that are not temporally disjoint, and observations that arrive out of temporal order.

The contributions of this paper include defining and implementing a framework meeting these criteria based on Markov Logic, a knowledge representation and reasoning system that combines first-order logic with probabilistic semantics. Our implementation includes a capable vision system for tracking the state of objects using an RGB-D (Kinect) camera together with an uncalibrated high-definition camera to increase accuracy. Low-level actions are defined in terms of qualitative spatial and temporal relations rather than visual appearance, so the system does not need to be trained on particular environments. We leverage a domain independent natural language parser to extract action descriptions and temporal constraints from the subject's narration. Our experiments demonstrate accurate recognition and tracking of complex plans, even as visual inputs to the system are purposefully degraded. Finally, we briefly describe how our future work on learning from demonstration builds upon this framework.

2. BACKGROUND & RELATED WORK

Our project builds upon work from a wide variety of fields: machine learning, knowledge representation, pervasive computing, computer vision, and computational linguistics. We provide a brief overview of only the most direct precedents.

Markov Logic [18] is a language for representing both logical and probabilistic information in the form of weighted logical formulas. Formulas that include quantified variables are taken to represent the set of ground formulas that can be formed by replacing the variables with constants. The probability of a possible world is proportional to the exponentiated sum of the weights of the ground formulas that are true in that world. The task of finding a most likely explanation of a set of data becomes maximum weighted satisfiability, and can be solved by local search or backtracking methods (*e.g.,* [3]).

Plan recognition was identified as a core reasoning problem in early research in AI and cognitive science [19]. Kautz (1991) developed a logical framework for plan recognition that met the criteria of expressiveness for action abstraction, decomposition, reification, and temporal generality, but did not handle probabilistic information and was never applied to observations from sensor data. The Markov Logic framework for plan recognition by Singla and Mooney (2011) handled probabilities, but was limited to a two-level hierarchy, did not reify actions, and was also never applied to sensor data.

Some groups have explored Markov Logic for activity recognition in video [11, 15, 23], but did not consider multi-level hierarchies and employed *ad hoc* rules for inferring unobserved events. Of these, Morariu *et al.* (2011) is closest to our framework, in that it associated events with time intervals rather than time points.

Lei *et al.* (2012) demonstrated robust tracking of low-level kitchen objects and activities (*e.g.,* pour, mix, *etc.*) using a consumer Microsoft Kinect-style depth camera (RGB-D). Their approach is similar to ours for low-level action recognition, but differs in that they inferred actions from object

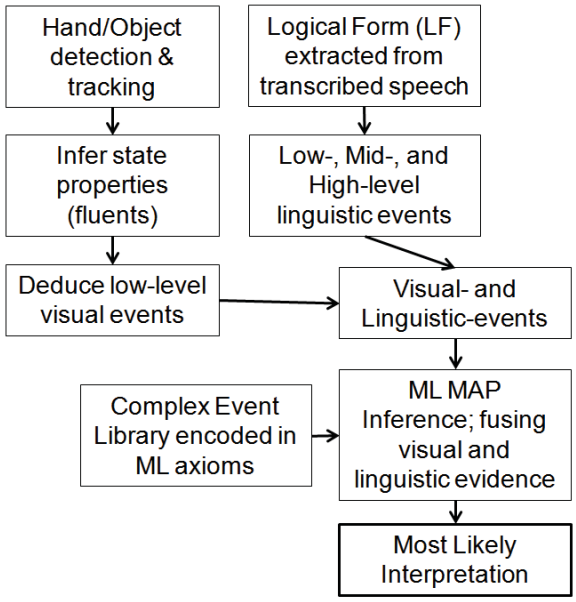

Figure 1: Overview of the multi-modal Markov logic (ML) framework for recognizing complex events. By obtaining visual and linguistic events through video and speech, our system provides a general framework for complex event recognition, fusing vision, language, and knowledge of complex event structure.

constraints and appearance-based motion flow, while we use object constraints and relative qualitative spatial position.

We employ the non-domain specific TRIPS parser [1] to extract action descriptions from narration. There is growing interest in machine learning and computational linguistics in models that unify visual perception and natural language processing. This includes using language to supervise machine vision (*e.g.,* [9]) and simultaneous learning of visual and linguistic attributes (color, shape, *etc.*) [13]. The grounded language approach of Tellex *et al.* (2011), like ours, integrates visual, linguistic, and background knowledge in a general probabilistic model, but has not yet considered plans or complex actions.

General formalisms such as stochastic grammars [14] and hierarchical hidden Markov models [6, 16] have been developed for representing and reasoning about hierarchically structured actions; however, grammars have difficulty representing non-disjoint actions, and HMM models fix the number of levels in the hierarchy. Others, such as event logic [5] provide a compact notation for probabilistic models relating interval-based actions and properties, while propagation networks [20] use partially-ordered plans to encode the transition relationships in a dynamic Bayesian network.

3. REPRESENTING COMPLEX EVENTS

We now describe a Markov Logic theory that meets our criteria for plan recognition. Throughout this section we will use the general term "event", rather than action or plan. We begin by introducing predicates that define the sets of event types, event instances, and relationships between events. Any particular domain is specified by defining the domains

Table 1: Core predicates in the event theory. Predicates that begin with "D" are used to define a domain, while Occurs, Part, Rel, Stime, and Etime hold for particular event instances.

Dabstracts(t_1, t_2) Event type t_1 abstracts type t_2.

Dpart(t_1, p, t_2) Events of type t_1 include a part p of type t_2.

DrelEP(t, r, p) The temporal relation r holds between any event of type t and its part p.

DrelPP(t, r, p_1, p_2) The temporal relation r holds between parts p_1 and p_2 of any instance of t.

Occurs(t, e) An event instance e of type t occurs.

Part(e_1, p, e_2) Event instance e_1 includes instance e_2 as a part p.

Rel(e_1, r, e_2) The temporal interval relation r holds between e_1 and e_2, which may be events or time intervals.

Stime(e, n) Event instance e starts at the integer-valued timestamp n.

Etime(e, n) Event instance e ends at the integer-valued timestamp n.

of these predicates. We then define *generic* axioms for predicting future and unobserved events on the basis of ongoing complex events, and abductively infer complex events from observations of subevents. This approach simplifies domain-specific knowledge engineering, and (in future work) turns the task of learning new events into learning the extent of the definitional predicates, rather than the unconstrained problem of learning arbitrary logical formulas.

Our implementation uses the implementation of Markov Logic called "Tuffy" [17]. Tuffy extends first-order syntax with scoping and datalog rules, which our implementation makes use of to substantially improve performance. Tuffy also restricts Markov Logic syntax by requiring that each formula be equivalent to a single clause. In order to keep this section brief and clear, however, we present logically equivalent axioms in pure Markov Logic.

Table 1 lists the predicates used to define a domain and to describe a particular situation in terms of the events that actually occur. Instances of events are reified, that is, are represented as individuals.

Abstraction Event types are organized into a hierarchy; an instance of a type is also an instance of all abstractions of the type. By default, an event of a given type is also an instance of some known specialization of the type. This is expressed by a weighted (soft) axiom. The weights (denoted by w) for soft rules can be learned from examples or estimated manually; in the experiments reported in this paper, estimated weights were sufficient. The axioms are thus:

$$\text{Dabstracts}(t_1, t_2) \wedge \text{Occurs}(e, t_2) \Rightarrow \text{Occurs}(e, t_1).$$

$$w: \ \text{Occurs}(e, t_1) \Rightarrow \exists\, t_2 \ \text{Dabstracts}(t_1, t_2) \wedge \text{Occurs}(e, t_2)$$

Figure 2: An example frame generated by the vision subsystem. The target objects are in labeled green bounding boxes, the subject's face is in a yellow bounding box, and her hands are outlined in cyan.

Temporal Temporal relationships between events are expressed using Allen's interval algebra [2], where event instances are treated as intervals. An integer timestamp can optionally be associated with the start and/or end time of an event. The intended semantics is captured by two sets of axioms, the first involving interval relations and endpoints, and the second involving triples of interval relations. An example of the first sort assert that if two events (intervals) meet, the end point of the first must equal the start point of the second; an example of the second is that "begun by" is transitive:

$$\text{Rel}(e_1, \text{Meets}, e_2) \wedge \text{Etime}(e_1, n_1) \Rightarrow \text{Stime}(e_2, n_2).$$

$$\text{Rel}(e_1, \text{BegunBy}, e_2) \wedge \text{Rel}(e_2, \text{BegunBy}, e_3) \Rightarrow \text{Rel}(e_1, \text{BegunBy}, e3).$$

For example, the formula

$$\text{Occurs}(\text{BoilWater}, \text{E}_1) \wedge \text{Part}(\text{E}_1, \text{Step}_1, \text{E}_2) \wedge \text{Occurs}(\text{FillKettle}, \text{E}_2) \wedge \text{Rel}(\text{E}_1, \text{BegunBy}, \text{E}_2) \wedge \text{Stime}(\text{E}_2, 109).$$

asserts that an instance of the complex event boiling water occurs, and that it is begun by the sub-event of filling a kettle. The filling starts at time 109. As a consequence of the general temporal axioms, the boiling water event also starts at time 109; both events end at unspecified times greater than 109.

Prediction Distinct from the abstraction hierarchy is a decomposition, or part-of, hierarchy. There are three types of axioms for complex events: prediction, constraint, and abduction. The *prediction* axiom assert that if a complex event occurs, each of its parts occurs by default.

$$w: \ \text{Occurs}(t_1, e_1) \wedge \text{Dpart}(t_1, p, t_2) \Rightarrow \exists\, e_2 \ \text{Occurs}(t_2, e_2) \wedge \text{Part}(e_1, p, e_2)$$

Constraint The *constraint* axioms assert that the defined temporal constraints among a complex event and its parts are satisfied.

$$\text{DrelEP}(t, r, p) \wedge \text{Occurs}(t, e) \wedge \\ \text{Occurs}(t_1, e_1) \wedge \text{Part}(e, p, e_1) \Rightarrow \text{Rel}(e, r, e_1).$$

$$\text{DrelPP}(t, r, p_1, p_2) \wedge \text{Occurs}(t, e) \wedge \\ \text{Occurs}(t_1, e_1) \wedge \text{Occurs}(t_2, e_2) \wedge \\ \text{Part}(e, p1, e_1) \wedge \mathit{Part}(e, p_2, e_2) \Rightarrow \text{Rel}(e_1, r, e_2).$$

Abduction Finally, *abduction* axioms allow complex events to be inferred on the basis of their parts. These axioms state that by default an event is part of a more complex event:

$$w: \ \text{Occurs}(t_1, e_1) \Rightarrow \\ \exists\, t_2\ e_2\ p \\ \text{Dpart}(t_2, p, t_1) \wedge \text{Occurs}(t_2, e_2) \wedge \text{Part}(e_2, p, e_1)$$

An observer should prefer more likely explanations and should not assume events occur without evidence. These preferences are captured by encoding a prior probability over the occurrence of events of various types by negative weighted clauses. For example,

$$-1\ \ \text{Occurs}(\text{MakeTea}, e)$$

$$-2\ \ \text{Occurs}(\text{MakeCocoa}, e)$$

indicates making tea is more likely than making cocoa. Specifically, if two worlds are identical except for the choice of tea or cocoa, then the odds ratio of the first world being true over the second is e^{-1}/e^{-2}.

4. DETECTING VISUAL EVENTS

Primitive visual events are inferred from interactions between the subjects' hands and objects in the environment. First, we detect and track the hands and objects from a RGB-D video on a frame-by-frame basis, and smooth the results to determine a set of time intervals over which hand and object related fluents (time-varying predicates) hold. Next, we look at where specific combinations of these fluents begin, end, or hold, and classify them as low-level visual events.

4.1 Hand and Object Detection and Tracking

While hands are difficult to detect on their own in video, faces can be reliably detected. The system employs face detection to first find the subjects face and determine skin color, and then uses this to aid in finding the hands. Objects and their orientation are detected and tracked by the combination of color and 3D bounding-boxes as computed from the point cloud. The following is a brief summary of the methods employed:

Skin modeling In order to make the vision subsystem adaptive to different lighting conditions, an image-specific Gaussian Mixture Model (GMM) is fitted over the pixels inside the detected face bounding box. Face detection is accomplished per frame by the Viola & Jones algorithm [24]. We assume that the majority of the area inside the detected face represents skin, which corresponds to the largest cluster in the fitted GMM. For a pixel outside the face bounding box, the Mahalanobis distance to the largest GMM component is computed as a skin score. In order to transform this real-valued score into a binary decision value, a two-parameter sigmoid classifier similar to Platt scaling in SVM (support vector machine) is trained on the fly.

Table 2: Set of visual fluents (time varying predicates) used in our scenarios.

Target	Property	Value
Single Object	Orientation	Straight, Tilted, Upside-down
	Motion	Stationary, In-motion
	Location	Counter, Sink, Cupboard
Object Pair	Relative Location	Above, Directly-above, Co-planar Below, Directly-below
	Distance	Adjacent, Near, Far
Object-Hand	Relation	Held, Not-held,
Subject	Location	Counter, Sink, Cupboard

Discriminative hand detection A discriminative Connected Components (CC) analysis is performed over the skin area binary map using SVM. For each CC in the skin area binary map, the following features are used: normalized CC size; normalized spatial distance to the detected face; width-height ratio of the CC bounding box; histogram of oriented gradients (HOG) [8]; and distance to the face.

Object identification Objects interacting with the hands are found by segmenting regions in the point cloud that are close to the hands but not part of the hands or body. Objects are differentiated from the hands by color and from the body by color and distance to the body. Objects on the table are found by subtracting the table and subject from the point cloud. Objects are identified using a nearest-neighbor match on color histogram and 3D bounding-box dimensions over a predefined set of named models. This simple approach to object identification was adequate for our experiments. For larger domains, the model library can be expanded to include various shape, texture, and other features.

Tracking and smoothing Since there are occlusion and perspective variations from time to time during the demonstration, object detection cannot be expected to be perfect. A multi-object tracking module is constructed to enforce temporal smoothing, particularly compensating for missed detections. Mean-shift-based tracking [7] is used for frame to frame object tracking, and color histogram distance is used as a matching score in the common Hungarian Algorithm to associate tracking and detections [4].

Fig. 2 shows a frame resulting from the vision subsystem, where hands and objects are detected, tracked and labeled. If skeleton tracking is readily available using the RGB-Depth cameras, we also use this information to identify and track the subject and their hands. This data is now used to infer low-level events.

Table 3: Examples of low-level events as defined by selected sets of fluents.

Low-level Event	Condition	Fluent: Target(s), Property and Value
Grasp	Begins	Object, Hand: Held
	Ends	Object, Motion: In-motion
	Holds	Object, Motion: Stationary
Release	Begins	Object, Motion: Stationary
	Ends	Object, Hand: Not-held
	Holds	Object, Hand: Held
Pour	Begins	$Object_1$, Orientation, Tilted
	Ends	$Object_1$, Orientation, Straight)
	Holds	$Object_1$, $Object_2$, Relative Location: Directly-above $Object_1$, Hand: Held

4.2 Low-Level Event Generation

The output of hand and object detection and tracking is a set of intervals over which visual fluents hold. As shown in Table 2, each fluent asserts that some object or pair of objects (first column) has a property (second column) with a particular discrete value (third column). Metric values from the visual system, such as the distance between two objects, are converted to discrete values, such as "Adjacent", "Near", or "Far", by simple threshholding. Discrete values for a fluent are mutually exclusive; for example, for the relative-location property, the value "Above" means the first object is on a higher plane than the second, but is not directly over the second; if it were, the value would be "Directly-above". Note that visual fluents are not tied to the particular domain of kitchen activities used in our experiments. The same (perhaps expanded) set of fluents would be useful for any domain that involves a person manipulating objects.

Next, primitive events, such as "Grasp", "Release", or "Pour", are generated in a deterministic fashion from changes in the fluent, as illustrated in Table 3. Fluents may trigger the beginning of an event, signal the end of the event, or be required to hold over the period of the event (but may in fact be longer than the event on either end). For example, a "Pour" event is begun by an object tilting, and ends when the object becomes straight. However, during the pour the object must be held and above some other receiving object. In addition to the fluent changes, the event definitions include type constraints on the objects involved (not shown in the table). For example, for a "Pour" event to be inferred, $Object_1$ must be of a type that can be poured (*e.g.,* a kettle) and $Object_2$ must be of a type that can be poured into (*e.g.,* a cup).

5. EXTRACTING EVENTS FROM LANGUAGE

The speech transcriptions from the audio are parsed with the TRIPS parser [1] to create a deep semantic representation of the language, the logical form (LF). Essential information regarding events are extracted from the LF via event extraction. The output from event extraction is converted into linguistic event instances and given as evidence to the inference system.

5.1 Logical Form

The TRIPS parser uses a semantic lexicon and ontology to create an LF that includes thematic roles, semantic types and semantic features, yielding richer representations than "sequence of words" models. To facilitate using language representation as features in activity recognition models, we added new semantic types in the ontology to correspond to objects and actions in the domain, such as "tea", "cup", or "steep". The new labels were usually specific subtypes of already existing semantic types. For example, the word "tea" was in the ontology under the general type TEAS-COCKTAILS-BLENDS, so we created the specific subtype TEA. This extension gives us greater transparency in the surface representation, but we retain the richness of the hierarchical structure and semantic features of our language ontology.

5.2 Event Extraction

The LFs are input to the TRIPS Interpretation Manager (IM), which computes crucial information for reasoning in the domain, including reference resolution. The IM extracts a concise event description from each clause, derived from each main verb and its arguments. The event descriptions are formulated in terms of the more abstract semantic types in the LF, resulting in short phrases such as CREATE TEA, CLOSE LID, and POUR WATER INTO CUP. By substituting the semantic information of referents for pronominal expressions, we derive more semantically meaningful descriptions. For example, the phrase "open it" would be described as OPEN LID in contexts where the pronoun "it" refers to the lid of the kettle. These phrases will be used as language features in our activity recognition models. Fig. 3 shows an example of an extraction from the LF for "Place tea bag in the cup." The objects "bag" and "cup" are identified as REFERENTIAL (*i.e.,* observable) by the IM, and the IM also includes the coreferential index for the first mention of the term.

```
(EXTRACTION-RESULT
 :VALUE ((EVENT V38801)
  (PUT V38801) (:THEME V38801 V38818)
  (:SHORT-DESCRIPTION V38801
   (PUT (:* BAG BAG) INTO CUP))
  (:INTO V38801 V38887)
  (:TENSE V38801 PRES)
  (REFERENTIAL V38818) (BAG V38818)
  (:ASSOC-WITH V38818 V38814)
  (:COREF V38818 V38185)
  (REFERENTIAL V38887) (CUP V38887)
  (:COREF V38887 V33594)
  (NONREFERENTIAL V38814) (TEA V38814))
  :WORDS (PUT THE TEA BAG INTO THE CUP))
```

Figure 3: Extraction for the utterance, "Place the tea bag in the cup."

Events described by language may have differing status with respect to the time of utterance. Our current system uses a decision tree method based on tense, aspects, modals, and event types to classify events according to their temporal relationship to the moment of utterance: past, ongoing, or future. In addition, since the current reasoning

is propositional, the system generates an atomic event description based on the short description produced. Putting this together, the analysis of the extraction in Fig. 3 produces a concise observation of form (PUTBAGINTOCUP :TEMPRELN FUTURE).

5.3 Encoding Language Events in Markov Logic

The final step in language processing encodes the processed LF in Markov logic. An event instance of the event type specified in the LF is generated, and the instance is asserted to occur. Any temporal attributes in the LF are converted to temporal constraints between the current clock time and the time of the event. For example, suppose the utterance leading to the form (PUTBAGINTOCUP :TEMPRELN FUTURE) is made during the time interval [190, 203]. A Markov logic formula of the following form is created, where E_{25} is the generated event instance:

$$\text{Occurs}(\text{PutBagIntoCup}, E_{25}) \wedge \text{Rel}(E_{25}, \text{After}, [190, 203])$$

A complication arises in the case where the event is a low-level type which could also occur as a visual event. Markov logic makes the assumption that different constants refer to different individuals. If the same event instance were both observed visually and described in language, it would be incorrect to use different event tokens for the visual and language-based observations. The case can be handled correctly by using an existentially quantified variable in the formula representing the language-based observation, rather than a constant, *e.g.,*

$$\exists\, e\ \ \text{Occurs}(\text{PutBagIntoCup}, e) \wedge \text{Rel}(e, \text{After}, [190, 203])$$

A new instance of the event type is added to the set of constants in the language, but the new instance is not explicitly asserted to occur. A consistent interpretation that does not make use of the new token will, in general, be preferred by the Markov logic inference engine over one that does, because as noted in Sec. 3, a negative weight is associated with the occurrence of each event.

6. EXPERIMENTS

We evaluate our framework on a multi-modal corpus collected from people conducting tasks in an instrumented kitchen, including making tea [22], making cocoa and making oatmeal. Participants were asked to conduct the activity and at the same time verbally describe the action being conducted. This section demonstrates how: (i) employing a complex event library improves visual event detection, and (ii) using both an event library and data from free-form spoken language can compensate for sparse visual input.

6.1 Data

Five sequences of making tea, and four sequences each of making cocoa and making oatmeal were chosen for evaluation. An RGB-Depth sensor, HD video camera, and microphones for recording speech were used for data collection. For ground truth, activities in the sessions were manually annotated by an expert observing recorded videos performed by the participants. Each low-level event in the video was annotated with an *action* (*e.g.,* grasp, carry, open), *attributes*, such as objects (*e.g.,* cup, kettle, teabox) and *paths* (*e.g.,* to, from, into).

Table 4: Example of a multi-level hierarchy of event types in our complex event library.

Top Level	Mid Level	Low Level
Make Tea	FillKettle	GraspKettle, CarryKettle, TurnonFaucet, FillWater TurnoffFaucet, CarryKettle ReleaseKettle
	GetIngredients	GoToCupboard, GetCupFromCupboard GetTeaboxFromCupboard
	PrepareTeabag	GraspTeabox, OpenTeabox, PutBagIntoCup
	BoilWater	TurnOnKettle, TurnOffKettle
	PourHotWater	GraspKettle, PourWaterIntoCup ReleaseKettle

We axiomatized the events that occurred in making tea into a multi-level hierarchy. The domain includes low-level events such as *GraspKettle*, mid-level complex events such as *BoilWater*, and general top-level events such as *MakeTea*. Table 4 lists the event types involved in an example of making tea. While not shown in the table, the *BoilWater* event abstracts two more specialized events: boiling water using an electric kettle, and boiling water using a microwave oven. Our complex event library also includes high-level events *MakeCocoa* and *MakeOatmeal*, which share many common mid- and low-level events with *MakeTea*.

6.2 Impact of the Complex Event Library

Low-level events, such as the ones shown in Table 4, are generated by fluents extracted from the vision subsystem. The low-level vision subsystem with the help of the RGB-D camera, detects the location of objects in the scene (*kettle, cup, teabox, etc.*), along with the locations of the subject and their hands in 3D space. The locations are quantified into scene-independent visual fluents, which serve as triggers that generate low-level events.

The left half of Table 5 (without complex event library) shows the raw low-level event detection performance for our 12 selected activity sessions divided into three top-level events. Approximately 2/3 of the events were detected on average. Some error counts were due to participants not being limited to a particular method of carrying on an activity, thus conducting actions that low-level detection was not able to either capture or detect accurately. However, despite having different people performing the same high level activity in different ways, a majority of low-level events were correctly identified. The differences over the top-level events are caused by objects used in the scenarios that are inheritently easier, or harder to detect than others.

We compare these raw recognition results to our complex event recognition framework using our event library. By defining the event type structure through the "D" predicates (via abstract, part and temporal relations) in Markov Logic, we were able to identify and match low-level events into the mid- and high-level event structure. The framework in many cases was able to fill in many of the missing events resulting in improved recall, while dismissing irrelevant events that were not part of the plan as being *unexplained* (*i.e.,* the cor-

Table 5: Recognition performance comparison for low-level events without and with using our complex event library. Performance improves when event hierarchy is taken into account.

Scenario	Without Complex Event Library				With Complex Event Library			
	# Correct Low-Level Events Detected	# Incorrect Low-Level Events Detected	Precision	Recall	# Correct Low-Level Events Detected	# Incorrect Low-Level Events Detected	Precision	Recall
Making Tea	59	32	0.65	0.66	72	11	0.87	0.81
Making Cocoa	64	40	0.62	0.89	68	18	0.79	0.94
Making Oatmeal	58	30	0.66	0.81	67	13	0.84	0.93
Total	181	102	0.64	0.77	207	42	0.83	0.89

Table 6: Number of linguistic events correctly and incorrectly interpreted extracted from narrations over all sessions of making tea.

	Low-level	Mid- and High-level
# Correct Events	10	8
# Incorrect Events	1	0

responding abduction axiom did not hold in the most likely interpretation), resulting in an improvement in precision. A comparison of results with and without using our complex event recognition framework is shown in Table 5.

6.3 Interpreting Linguistic Events

The subject's utterances from the 5 making tea sessions were transcribed, and the text parsed and put into an initial logical form by the TRIPS parser. Events were extracted according to the method described in Section 5. The processed LF and its temporal constraints are extracted and encoded in Markov Logic. Lingustic events that were not represented by any event in our library was considered irrelevant and discarded. We call events originating from language *linguistic events* to differentiate it from events generated using visual data.

While a total of 89 linguistic events were generated from the speech data from our event corpus, only 19 were identified as relevant to our activities, and the remaining events were filtered out. Table 6 shows the number of linguistic events correctly and incorrectly identified by the system. A majority of the events from language confirmed evidence already recognized by the visual system. Out of the 10 correctly identified low-level linguistic events, 7 explained events that were consistent with the visual events, while 3 were events not previously identified from visual data.

An advantage linguistic events have over visual events is that they can directly describe mid- and high-level events, which can be interpreted as corresponding predicates in Markov Logic without much difficulty. A purely vision-based system on the other hand must rely on using the event structure to infer higher-level events. During our evaluations, 8 mid- and high-level events were explicitly identified through language.

6.4 Impact of Language in Sparse Visual Data

Linguistic input is most important when the visual input is noisy and incomplete, and/or the task is plan learning rather than plan tracking. As noted, this paper assumes the plan library is known, while our future work will take on the learning case. The performance of the vision system in our testing kitchen was high enough that only minor further improvement could be obtained by integrating language. Therefore, in order to fully explore the power of language to aid event tracking in the face of sparse visual data, we ran a series of experiments where we deliberately degraded the input to the Markov Logic engine from the vision system. We varied the input from 100% to only 25% of the visually detected low-level events.

Table 7: Impact of language with varying amounts of visual event data. Precison and recall are averaged over all sessions of making tea.

% Visual Events	Without Linguistic Events		With Linguistic Events	
	Precision	Recall	Precision	Recall
100	0.87	0.80	0.80	0.92
75	0.89	0.78	0.86	0.93
50	0.98	0.43	0.94	0.60
25	0.90	0.21	0.98	0.60

There are many situations where visual data is similarly sparse. For example, an object may be too small to recognize, or occlusion of the subject or objects may cause catastrophic failure in the visual tracking system. In such cases, language aids the system in recognizing events and plans. As shown in Table 7, in the absence of linguistic events, recall of low-level events degrades linearly with the amount of visual event data. However, the use of linguistic events, especially mid- and high-level events used in conjunction with sparse visual data significantly improves overall recognition recall.

7. CONCLUSIONS

In this paper we have presented a robust Markov Logic framework for recognizing hierarchical events from a complex event library. We have described a Markov Logic theory that is able to represent complex events, and have used this theory to implement a system that can recognize kitchen activities in a real world setting, combining visual and linguistic evidence.

In highly structured domains, formulating and applying complex event knowledge results in an improvement of event recognition in the form of contextual information. We have

examined the advantages of using multimodal data through a Markov logic framework that can directly integrate multiple modes using visual and linguistic events. We have shown the usefulness of linguistic events as additional input to our system in the absence of visual events. To the best of our knowledge, our system is the first general plan recognition system that can recognize and trace arbitrarily complex events on the basis of both visual and linguistic input.

Our current system is only the first step in our larger project of creating a system that can learn new complex activities from demonstration. The use of multimodal data, such as vision and language, introduces the possibility of learning new complex activities in a rich and dynamic manner. The flexibility of Markov Logic provides a compact but robust method of representing visual and linguistic events. For future work, we plan on formalizing activity learning as the task of extending the domains of the event definition predicates so as to reduce the overall cost (*i.e.,* increasing the probability) of the observed demonstration.

8. ACKNOWLEDGMENTS

This work was supported by NSF Awards IIS-1012017, IIS-1012205 and N00014-11-10417, ARO Award W911NF-08-1-0242, DoD SBIR Award N00014-12-C-0263, the Google Faculty Research Award, and the Intel Science & Technology Center for Pervasive Computing (ISTC-PC).

9. ADDITIONAL AUTHORS

Ce Zhang (Department of Computer Sciences, University of Wisconsin-Madison, email: `czhang@cs.wisc.edu`).

10. REFERENCES

[1] J. Allen, M. Swift, and W. de Beaumont. Deep semantic analysis of text. In *Proc. Semantics in Text Processing*, STEP '08, pages 343–354, 2008.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, Nov. 1983.

[3] C. Ansótegui, M. L. Bonet, and J. Levy. Sat-based maxsat algorithms. *Artifical Intelligence*, 196, 2013.

[4] S. Blackman. *Multiple-target tracking with radar applications*. Artech House radar library. Artech House, 1986.

[5] W. Brendel, A. Fern, and S. Todorovic. Probabilistic event logic for interval-based event recognition. In *24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 3329–3336, 2011.

[6] H. H. Bui. A general model for online probabilistic plan recognition. In *Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003)*, 2003.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, May 2003.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[9] S. Gupta and R. J. Mooney. Using closed captions as supervision for video activity recognition. In *Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2010)*, 2010.

[10] H. Kautz. A formal theory of plan recognition and its implementation. In J. Allen, H. Kautz, R. Pelavin, and J. Tennenberg, editors, *Reasoning About Plans*, pages 69–126. Morgan Kaufmann Publishers, 1991.

[11] A. Kembhavi, T. Yeh, and L. Davis. Why did the person cross the road (there)? scene understanding using probabilistic logic models and common sense reasoning. In *11th European Conference on Computer Vision (EECV 2010)*, 2010.

[12] J. Lei, X. Ren, and D. Fox. Fine-grained kitchen activity recognition using rgb-d. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*, pages 208–211, 2012.

[13] C. Matuszek, N. FitzGerald, L. S. Zettlemoyer, L. Bo, and D. Fox. A joint model of language and perception for grounded attribute learning. In *29th International Conference on Machine Learning (ICML 2012)*, 2012.

[14] D. Moore and I. Essa. Recognizing multitasked activities using stochastic context-free grammar. In *In Proceedings of AAAI Conference*, 2001.

[15] V. I. Morariu and L. S. Davis. Multi-agent event recognition in structured scenarios. In *24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, 2011.

[16] S. Natarajan, H. H. Bui, P. Tadepalli, K. Kersting, and W. Wong. Logical hierarchical hidden Markov models for modeling user activities. In *In Proc. of ILP-08*, 2008.

[17] F. Niu, C. Ré, A. Doan, and J. W. Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *Proceedings of the VLDB Endowment (PVLDB)*, 4(6):373–384, 2011.

[18] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

[19] C. F. Schmidt, N. S. Sridharan, and J. L. Goodson. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artifical Intelligence*, 11(1-2), 1978.

[20] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation Networks for Recognition of Partially Ordered Sequential Action. In *Proceedings of IEEE CVPR04*, 2004.

[21] P. Singla and R. J. Mooney. Abductive Markov Logic for plan recognition. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*, 2011.

[22] M. Swift, G. Ferguson, L. Galescu, Y. Chu, C. Harman, H. Jung, I. Perera, Y. Song, J. Allen, and H. Kautz. A multimodal corpus for integrated language and action. In *Proc. of the Int. Workshop on MultiModal Corpora for Machine Learning*, 2012.

[23] S. Tran and L. Davis. Visual event modeling and recognition using markov logic networks. In *10th European Conference on Computer Vision (EECV 2008)*, 2008.

[24] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.