

# Joint Video and Text Parsing for Understanding Events and Answering Queries

Kewei Tu, Meng Meng, Mun Wai Lee, Tae Eun Choe, Song-Chun Zhu

**Abstract**—We propose a multimedia analysis framework to process video and text jointly for understanding events and answering user queries. Our framework produces a parse graph that represents the compositional structures of spatial information (objects and scenes), temporal information (actions and events) and causal information (causalities between events and fluents) in the video and text. The knowledge representation of our framework is based on a spatial-temporal-causal And-Or graph (S/T/C-AOG), which jointly models possible hierarchical compositions of objects, scenes and events as well as their interactions and mutual contexts, and specifies the prior probabilistic distribution of the parse graphs. We present a probabilistic generative model for joint parsing that captures the relations between the input video/text, their corresponding parse graphs and the joint parse graph. Based on the probabilistic model, we propose a joint parsing system consisting of three modules: video parsing, text parsing and joint inference. Video parsing and text parsing produce two parse graphs from the input video and text respectively. The joint inference module produces a joint parse graph by performing matching, deduction and revision on the video and text parse graphs. The proposed framework has the following objectives: Firstly, we aim at deep semantic parsing of video and text that goes beyond the traditional bag-of-words approaches; Secondly, we perform parsing and reasoning across the spatial, temporal and causal dimensions based on the joint S/T/C-AOG representation; Thirdly, we show that deep joint parsing facilitates subsequent applications such as generating narrative text descriptions and answering queries in the forms of who, what, when, where and why. We empirically evaluated our system based on comparison against ground-truth as well as accuracy of query answering and obtained satisfactory results.

**Index Terms**—Joint video and text parsing; Knowledge representation; And-Or graph; Multi-media video analysis; Query answering.



## 1 INTRODUCTION

### 1.1 Motivation and objective

Video understanding is an important aspect of multimedia content analysis that aims at automated recognition of the objects, scenes and events in the video. It is a crucial component in systems that improve human-computer interaction, such as (a) automatic text generation from videos to facilitate search and retrieval of visual information from the web, and (b) surveillance systems that answer human queries of who, what, when, where and why. However, understanding contents from video alone has been found to be challenging because of factors such as low resolution, deformation and occlusion; in addition, some aspects of objects and events are hidden, for example, a person’s intentions and goals.

A significant portion of videos are accompanied by textual descriptions, which provide supplementary information for video understanding. Figure 1 shows three examples, with decreasing levels of overlap between the video contents and the text contents. (a) Movies or television shows and their screenplays, which typically have the largest overlap. (b) Surveillance videos and human intelligence in the form of narrative text descriptions of the scenes and events, which are usually partially overlapped. (c) News report footages and their closed

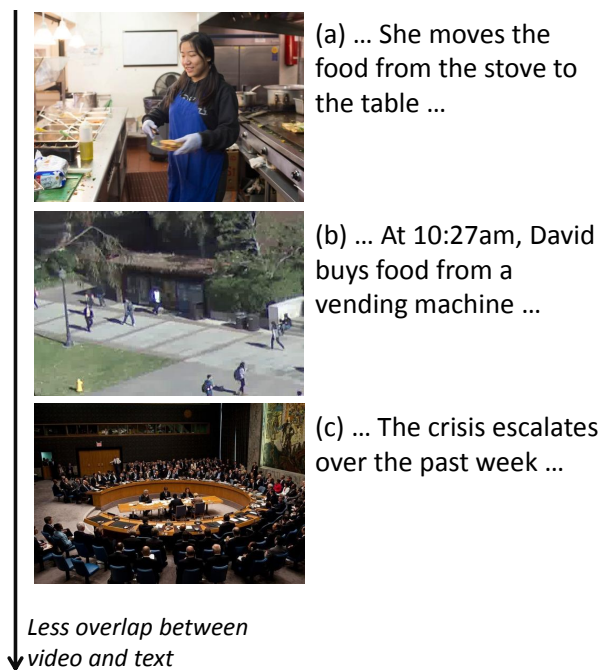


Fig. 1. Examples of videos accompanied with texts. (a) A television show footage and its screenplay; (b) A surveillance video and the human intelligence descriptions; (c) A news report footage and its closed captions.

*K. Tu, M. Meng and S.C. Zhu are with the Department of Statistics, University of California, Los Angeles.*

*M.W. Lee is with Intelligent Automation, Inc.*

*T.E. Choe is with ObjectVideo Inc.*

captions of the news anchor’s narrative as well as a few lines of on-screen text summarizing the news, which often have the least overlap.

There have been growing interests in multimedia research to process video and text jointly and therefore integrate vision and language – the two major sensory modalities for human intelligence. On the one hand, vision provides rich sensory signals to ground language symbols denoting objects, scenes and events; it supplements language with low-level details that are typically not mentioned in text descriptions. On the other hand, language conveys semantic concepts and relations underlying the visual world, some of which may be hidden or obscure from the visual channel (such as names, certain attributes, functions, intents, and causality). Furthermore, since vision and language provide information of the same scene from two different channels, they could disambiguate and reinforce each other and produce a more accurate joint interpretation.

In contrast to the bag-of-words representation widely used in the video understanding and multimedia literature (e.g., [1], [2], [3], [4]), in this work we propose to represent the joint interpretation of video and text in the form of a parse graph. A parse graph is a labeled directed graph that can be seen as an extension of the constituency-based parse tree used in natural language syntactic parsing [5]. It has been employed in computer vision to model objects [6], scenes [7], [8] and events [9], [10]. A node in a parse graph represents an entity that can be an object, an event or a status of an object (i.e., fluent). An edge in a parse graph represents a relation between two entities. Together they represent the compositional structures of spatial information (for objects and scenes), temporal information (for actions and events) and causal information (causalities between events and object statuses). Such an explicit semantic representation facilitates subsequent applications such as text generation and query answering.

In order to handle different levels of overlap in the video and text, joint parsing must overcome the following challenges.

- When overlap exists between video and text, we need to solve the coreference problem, i.e., identifying the correspondence between the entities and relations observed in the video and those described in the text. For example, we may need to identify that a person detected in the video and a name mentioned in the text correspond to the same person. This problem can be complicated when multiple entities or relations of the same type appear in the same scene.
- When there is little overlap between video and text, we need to fill in additional information to construct a coherent and comprehensive joint interpretation. For example, if the video shows a person waiting at a food truck and the text mentions a person buying food, then we need to infer that waiting at a food truck is typically followed by buying food in order

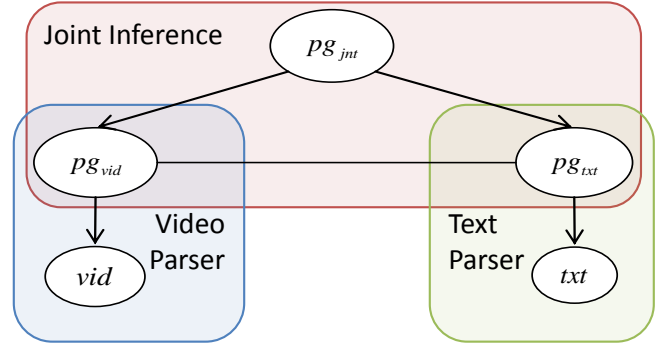


Fig. 2. The graphical model for joint parsing of video and text. See text for details.

to connect the video and text.

- Occasionally, the content detected from the video may conflict with the content of the text, because of either erroneous parsing or inaccurate input. We need to resolve the conflicts and provide a consistent joint interpretation.

## 1.2 Overview of our work

We focus on joint parsing of surveillance videos and narrative text descriptions, but our framework can be employed or extended to process other types of video-text data as well. Figure 2 shows our probabilistic generative model. Let  $vid$  and  $txt$  denote the input video and text respectively,  $pg_{vid}$  the parse graph that interprets the input video,  $pg_{txt}$  the parse graph that interprets the input text, and  $pg_{jnt}$  the joint parse graph for both the video and the text. Our goal can be formulated as optimizing the posterior probability of the three parse graphs  $pg_{jnt}$ ,  $pg_{vid}$  and  $pg_{txt}$  given the input video  $vid$  and text  $txt$ .

$$P(pg_{jnt}, pg_{vid}, pg_{txt} | vid, txt) \propto P(pg_{jnt})P(pg_{vid}, pg_{txt} | pg_{jnt})P(vid | pg_{vid})P(txt | pg_{txt}) \quad (1)$$

Our system for optimizing this posterior probability can be divided into three modules. First, a *video parser* generating candidate video parse graphs  $pg_{vid}$  from the input video  $vid$ . The video parser encodes the conditional probability  $P(vid | pg_{vid})$ . It is also guided by a spatial-temporal-causal And-Or graph (S/T/C-AOG) [6], [7], [9], [10], which models the compositional structures of objects, scenes and events. Second, a *text semantic parser* parsing the text descriptions  $txt$  into the text parse graph  $pg_{txt}$ . We build our semantic parser on top of the Stanford Lexicalized Parser [11], focusing on parsing narrative event descriptions to extract information about objects, events, fluent changes and the relations between these entities. The text parser specifies the conditional probability  $P(txt | pg_{txt})$ . Third, a *joint inference module* producing a joint parse graph  $pg_{jnt}$  based on the video and text parse graphs produced by the first two modules. The joint inference takes into account 1) the relationships between the joint parse graph and the video and text

parse graphs, which are encoded in the conditional probability  $P(pg_{vid}, pg_{txt} | pg_{jnt})$ , and 2) the compatibility of the joint parse graph with the background knowledge represented in the S/T/C-AOG, as encoded in the prior probability  $P(pg_{jnt})$ . We employ three types of operators in constructing the joint parse graph:

- Solving the coreference problem by *matching* the video and text parse graphs
- Filling in the potential gap between video and text by *deduction*.
- Resolving possible conflicts via *revisions* to the video and text parse graphs.

Figure 3 and 4 show two examples of joint parsing from videos and texts with different levels of overlap. In each parse graph shown in the figures, we use different shapes to represent objects, events and fluents, and use different edge colors to represent different relations; the label on a node or edge indicates the semantic type of the denoted entity or relation. For each event node, the `Agent` relation connects it to the action initiator, and the `Patient` relation connects it to the action target. The set of event nodes are laid out horizontally based on their occurring time and vertically based on their compositional complexity. For a fluent node, we use levels to indicate its value at each time point. Figure 3(a) shows a surveillance video and the human intelligence description of a road scene, in which the video and text cover the same event and have significant overlap. Figure 3(b) shows the results of video parsing, text parsing and joint inference. The video and text parses reinforce each other in the joint parse via overlapping (e.g., the stopping event, the building), and they also supplement each other by providing additional information (e.g., the taking-picture event is too subtle to recognize from the video but is provided in the text). Figure 4(a) shows a surveillance video and text description of a courtyard scene. The video shows a few people waiting and taking turns, but their target is not recognized by the video parser; the text only mentions a food truck. The video and text parses have no direct overlap in terms of object entities, but these objects are involved in a common food purchase event. Figure 4(b) shows the results of video parsing, text parsing and joint inference. The joint parse graph identifies the food truck as the target of people’s actions and infers the types of the events as well as the fluent changes resulting from the events, thus providing a more coherent and comprehensive interpretation of the scene.

We encode the prior knowledge of parse graphs in the S/T/C-AOG and use it to guide both video parsing and joint inference, which is another contribution of this work. An AOG is an extension of a constituency grammar used in natural language parsing [5] to represent hierarchical compositions of objects, scenes and events. An AOG has a hierarchical structure with alternating layers of And-nodes and Or-nodes. An And-node represents the configuration of a set of sub-entities to form

a composite entity (e.g., waiting followed by ordering composes a buying event). An Or-node represents the set of alternative compositional configurations of an entity (e.g., a buying event can be either “waiting and ordering” or “ordering without waiting”).

- A spatial And-Or graph (S-AOG) [6], [7] models the spatial decompositions of objects and scenes.
- A temporal And-Or graph (T-AOG) [9] models the temporal decompositions of events to sub-events and atomic actions.
- A causal And-Or graph (C-AOG) [10] models the causal decompositions of events and fluent changes.

These three types of AOGs are interconnected and together they form a joint S/T/C-AOG. A parse graph is an instance of an AOG that selects only one of the alternative configurations at each Or-node of the AOG. The prior probability of a parse graph is specified by the energy of configuration selections made at the Or-nodes of the AOG as well as the energy of the compositional configurations at the And-nodes of the AOG contained in the parse graph.

### 1.3 Evaluation

To empirically evaluate our system, we collected two datasets, each containing surveillance videos of daily activities in indoor or outdoor scenes and a set of text descriptions provided by five human subjects. We asked each human subject to provide text descriptions at three levels of details, so we can study how the amount of text descriptions and their degrees of overlap with the video content can impact the performance of joint parsing. The joint parse graphs produced by our system were then evaluated using two different approaches.

In the first approach, we compare the joint parse graphs against the ground truth parse graphs which were constructed based on the merged parses of all the text descriptions from all the human subjects. We measure the precision and recall where precision is the percentage of the joint parse graph that is correct (i.e., contained in the ground truth) and recall is the percentage of the ground truth that is covered by the joint parse graph.

One important application of video understanding is to answer queries from human users. Therefore, in the second evaluation approach, we measure the accuracy of a query answering system based on the joint parse graph produced by our system. We collected a set of natural language queries that mimics the natural human-computer interaction and includes questions in the forms of who, what, when, where and why. The answers produced by the system were then compared against the correct answers provided by human subjects.

The evaluation results show that the joint parse graphs provide more accurate and comprehensive interpretations of the scenes than the video and text parse graphs. In addition, we find that additional text input improves the joint parse graphs but with diminishing return.

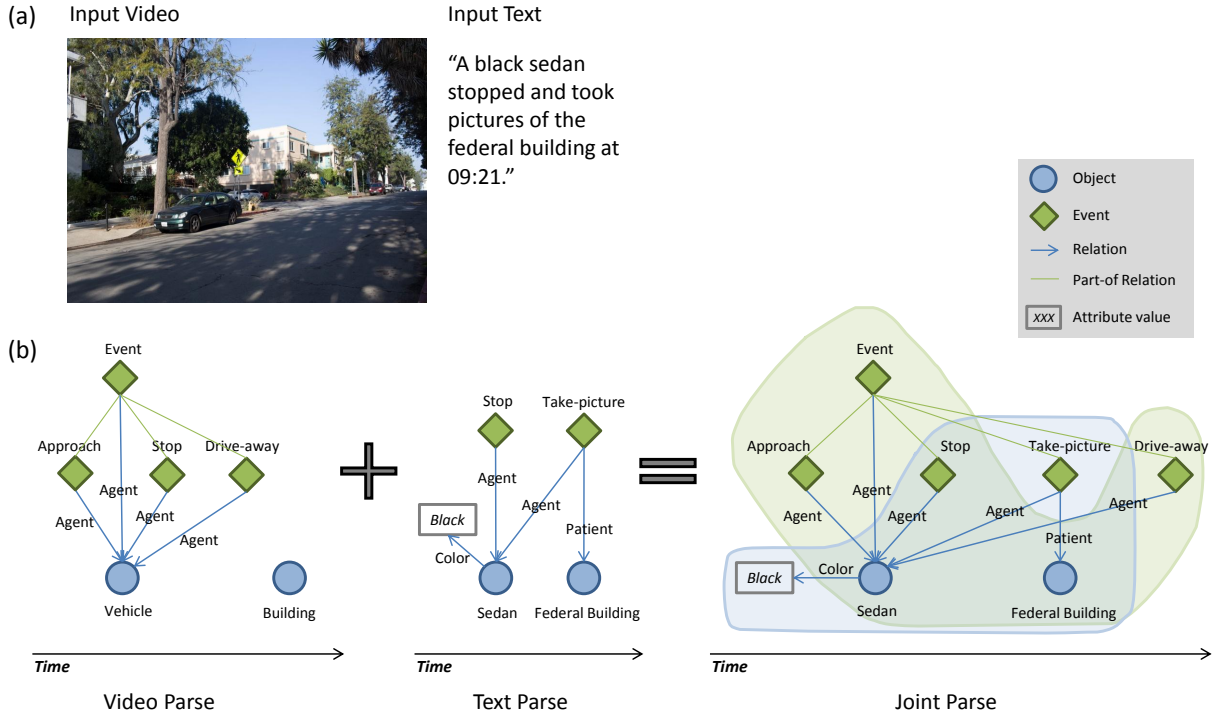


Fig. 3. (a) An example surveillance video and the text descriptions of a parking lot scene. (b) The video parse graph, text parse graph and joint parse graph. The video and text parses have significant overlap. In the joint parse graph, the green shadow denotes the subgraph from the video and the blue shadow denotes the subgraph from the text.

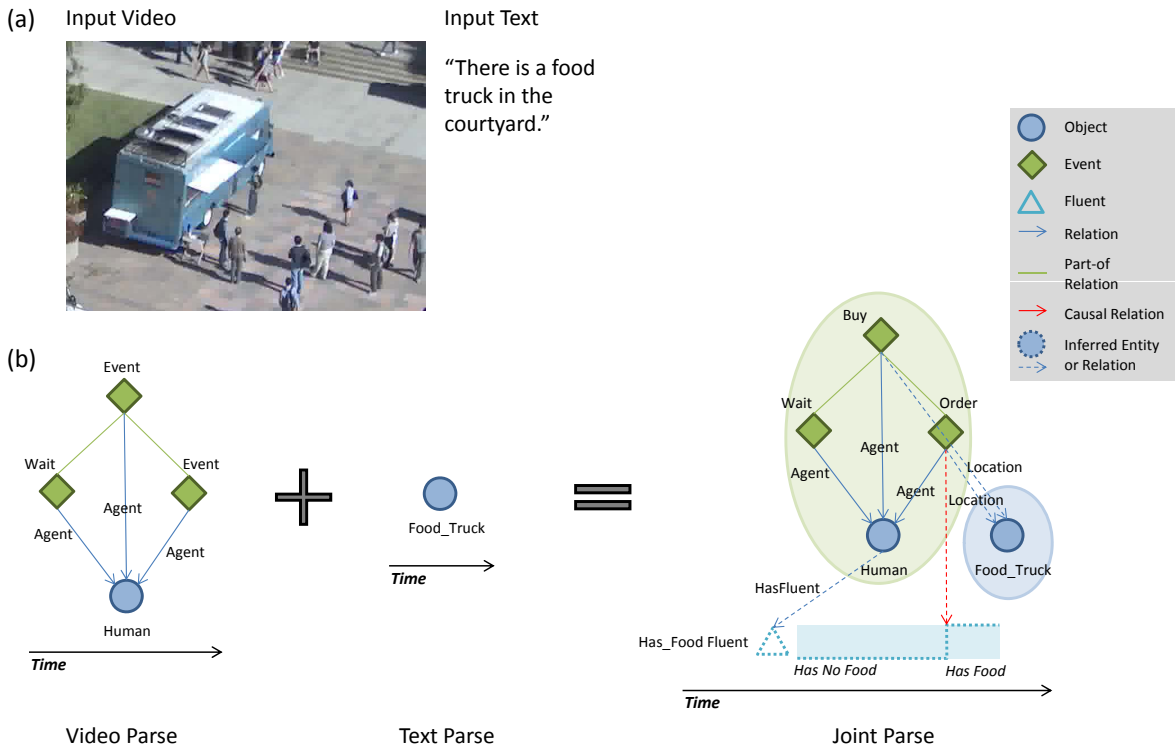


Fig. 4. (a) An example surveillance video and the text descriptions of a courtyard scene. (b) The video parse graph, text parse graph and joint parse graph. The video and text parses have no overlap. In the joint parse graph, the green shadow denotes the subgraph from the video and the blue shadow denotes the subgraph from the text.

## 1.4 Related work

### 1.4.1 Bag-of-words based approaches

There has been a large body of work in the areas of multimedia, computer vision and machine learning on joint processing of video/image and text. The majority of the existing work is based on the bag-of-words (BoW) representation for both video/image and text. In the BoW representation, a text is represented as a set of unordered words; and a video/image is represented as a set of unordered visual words, each of which is a feature vector (e.g., SIFT and HoG) of an image patch. The BoW representation is easy to extract and process, but it ignores the compositional structures based on the words. Therefore, BoW-based approaches typically perform coarse-level processing of the video/image and text, e.g., classification or retrieval, instead of producing a full interpretation of the input video/image and text. For example, for joint video and text processing, Feng et al. [1] applied a joint model of video/image and text for both automatic image annotation and retrieval; Iyengar et al. [12] modeled videos and text jointly for the purpose of multimedia information retrieval; Yang et al. [13] addressed the problem of video classification by joint modeling of videos and transcript texts; Xu et al. [14] performed sports video semantic event detection based on joint analysis of videos and webcast texts. For joint processing of image and text, Paek et al. [15] performed scene classification with a TF-IDF based approach applied to both images and the accompanying text; Barnard et al. [2] and Blei [16] discussed a series of probabilistic models to capture the dependencies between image elements and words; Berg and Forsyth [17] demonstrated a method for identifying images containing categories of animals based on word and visual cues; Monay and Gatica-Perez [3] applied the probabilistic latent semantic analysis model for annotated images to improve automatic image indexing; Wang et al. [4] simultaneously modeled the image elements, text annotations and image class labels using a probabilistic topic model; Liu et al. [18] presented an image retrieval system by leveraging large-scale web image and their associated textual descriptions; Jia et al. [19] proposed a probabilistic model that encodes relations between loosely related images and text descriptions; Feng and Lapata [20] learned a probabilistic topic model of visual and textual words for the purpose of automatic image caption generation. In contrast to the existing work, our work aims at deep semantic parsing of video and text, i.e., we try to identify the compositional structures contained in the video and text, based on which we extract and explicitly represent a full interpretation of the input video and text. Our scope of parsing includes spatial parsing (of objects and scenes), temporal parsing (of actions and events) and causal parsing (of events and fluents), some of which have rarely been explored in the previous work of joint processing of video/image and text.

### 1.4.2 Parsing based approaches

Our work is built on the recent advancement on image/video parsing, which identifies the hierarchical compositional structures contained in the image or video. In particular, spatial parsing identifies the spatial compositional structures of objects and scenes [21], [6], [22], [23], [24], [25], [7], [8]; temporal parsing recognizes the temporal compositional structures of events [26], [27], [28], [29], [30], [31], [9], [32]; and causal parsing detects the causal relations between events and fluent changes [33], [34], [10].

Our work also involves semantic parsing of text descriptions, which is related to a variety of semantic parsing approaches studied in the natural language processing community, for example, a supervised approach [35], an unsupervised approach [36], and a dependency based approach [37]. Semantic text parsing is typically built on top of a syntactic parser such as the Stanford Lexicalized Parser [11].

For joint parsing of video and text, Hobbs and Mulkar-Mehta [38] described a preliminary framework that interprets the combination of video and text from news broadcasts, which is the work most closely related to ours. Compared with their work, our system extracts information directly from the raw data of video and text, and relies on a probabilistic generative model to handle uncertainty; besides, we have done comprehensive experiments to evaluate our system. The graph-based referential grounding approach proposed by Liu et al. [39] is also related to our work, but they focused on spatial relations between objects while we study spatial-temporal-causal parsing of objects, scenes and events.

## 1.5 Contributions

The contributions of our paper include three aspects:

- Although there has been a lot of previous work in video parsing and text semantic parsing, our work is the first to jointly parse video and text for more accurate and comprehensive parsing results. Our work of joint semantic parsing also goes beyond the traditional bag-of-words approaches to joint processing of video and text.
- Although spatial, temporal and causal AOGs and their parsing techniques have been separately investigated before, our work is the first to propose the joint S/T/C-AOG which crosses the spatial, temporal and causal dimensions and enables the propagation of information between them during parsing.
- We propose novel evaluation approaches of video understanding based on parse graph comparison and query answering through storylines. In particular, the query-answering-based evaluation mimics the natural human-computer interaction via questions of who, what, when, where and why, which goes beyond the conventional evaluation frame-

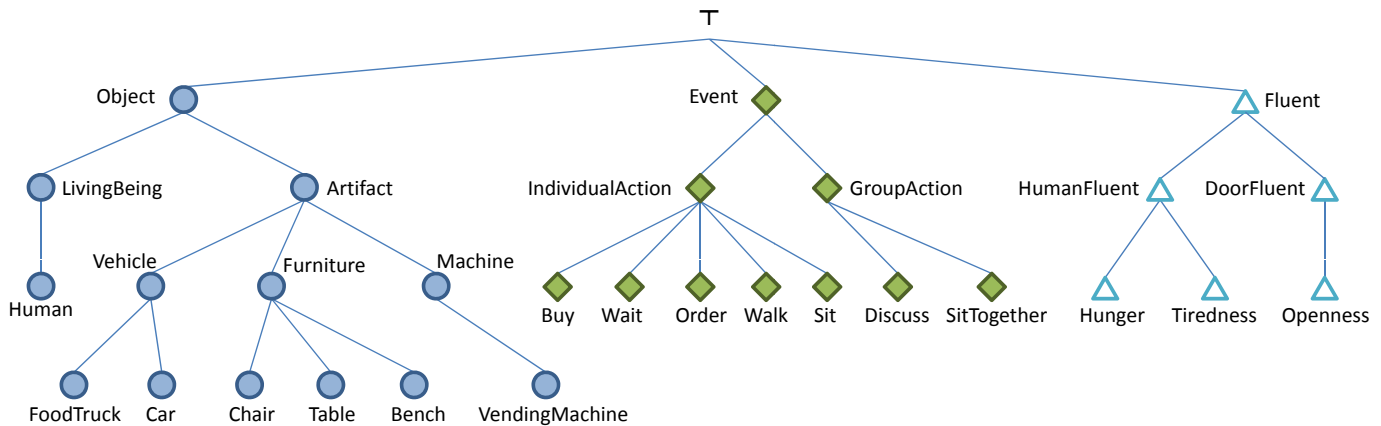


Fig. 5. The taxonomy in an example ontology for the courtyard scene. The symbol  $\top$  at the root node represents the *universal* type, which is a supertype of every other type.

works such as those based on classification or detection.

## 1.6 Outline of the paper

The rest of the paper is organized as follows. Section 2 introduces our knowledge representations. Section 3 and 4 present the approaches of video parsing and text parsing respectively. Section 5 introduces our joint inference module. Section 6 introduces the application of joint parsing in natural language query answering. Section 7 presents our experiments. Section 8 concludes the paper.

## 2 REPRESENTATIONS

### 2.1 Ontology

An ontology is a “formal, explicit specification of a shared conceptualisation” [40]. We use an ontology to specify the types of entities and relations that may appear in the parse graphs and therefore define the scope of our study. This ontology can be manually constructed using an ontology editor (e.g., [41], [42], [43]), adapted from an existing ontology (e.g., [44], [45]), or automatically or semi-automatically learned from data (e.g., [46], [47], [48]).

As typical for an ontology, we organize the types of entities (i.e., concepts) into a taxonomy. A taxonomy is a directed acyclic graph in which a directed edge between two concepts denotes a supertype-subtype relationship. As will be shown in section 5, we rely on the taxonomy to compute semantic distances between concepts when doing joint inference. We divide the concepts into three main categories at the top level of the taxonomy.

- Object, which represents any physical objects including human.
- Event, which includes any basic actions (e.g., sitting and walking) as well as any activities that are composed of a sequence of actions (e.g., buying).
- Fluent, which is used in the AI community to refer to an attribute of a physical object that can change

over time [49], e.g., the openness attribute of a door and the hunger attribute of a human. The change of fluents helps detect or explain human actions in a scene and thus is important in joint parsing.

Figure 5 shows an example taxonomy for the courtyard scene.

We also specify a set of relations in the ontology. These relations can be organized into a relation taxonomy, allowing the computation of semantic distances between relations. A small set of core relations are introduced in the next section.

### 2.2 Knowledge representation by And-Or graphs

We represent the compositional structures of objects, scenes and events using a spatial-temporal-causal And-Or graph (S/T/C-AOG). We will define probabilistic models on the AOG to address the stochasticity of the compositional structures in section 2.4. An AOG is an extension of a constituency grammar used in natural language parsing [5]. Like a constituency grammar, an AOG represents possible hierarchical compositions of a set of entities. However, an AOG differs from a constituency grammar in that it contains additional annotations and relations over the entities. More specifically, an AOG consists of a hierarchical structure with alternating layers of And-nodes and Or-nodes. An And-node represents the decomposition of an entity of a specific type into a set of sub-entities. A *Part-of* relation is established between each sub-entity and the composite entity. The And-node also specifies a set of relations between the sub-entities, which configure how these sub-entities form the composite entity. An Or-node represents the alternative configurations of an entity of a specific type. AOGs has been employed in computer vision to model the hierarchical decompositions of objects [6], scenes [7], [8] and events [9], [10]. In this work, we use an AOG to model objects, scenes, events and perceptual causal effects in the joint interpretation of video and text.

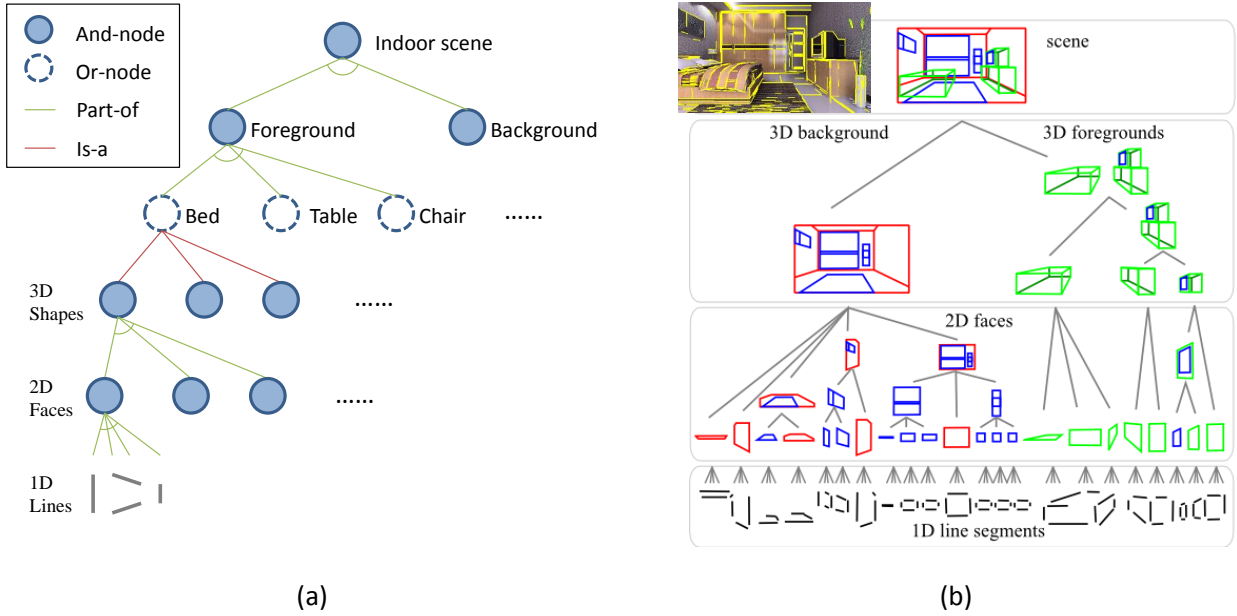


Fig. 6. (a) An example spatial And-Or graph (S-AOG). Each node represents a type of scenes, objects or their parts; an And-node represents the spatial composition of its child nodes; an Or-node represents a selection among its child nodes; a leaf node represents an atomic object. (b) An example spatial parse graph from [7], which is a realization of the S-AOG by making selections at Or-nodes.

### 2.2.1 Spatial And-Or graph

A spatial And-Or graph (S-AOG) [6], [7] models the spatial decompositions of objects and scenes. Figure 6(a) shows an example S-AOG for indoor scenes. Each node in the S-AOG represents a type of scenes, objects or their parts. An And-node in the S-AOG represents a scene or object that is the spatial composition of a set of parts. For example, a table can be the spatial composition of a table top above four legs; and the background of an indoor scene can be the spatial composition of multiple 2D planes (i.e., the walls, floor and ceiling) that are hinged. An Or-node in the S-AOG represents alternative configurations of a scene or object. For example, a table can have many different styles; and an indoor scene may have a few different typical viewpoints. A leaf node in the S-AOG represents an atomic object that cannot be further decomposed.

### 2.2.2 Temporal And-Or graph

A temporal And-Or graph (T-AOG) [9] models the hierarchical decompositions from events to sub-events and then to atomic actions. Figure 7(a) shows an example T-AOG. Each node in the T-AOG represents a type of event or action occurred in a time interval. An And-node in the T-AOG represents an event that is the temporal composition of a set of sub-events. For example, a buying event can be the temporal composition of waiting followed by ordering. An Or-node in the T-AOG represents alternative configurations of an event, e.g., a buying event can be either “waiting and ordering” or “ordering without waiting”. A leaf node in the T-AOG represents

an atomic action which is defined by properties of the action initiator(s) and the action target(s) (e.g., the pose of a human, the state of an object) as well as the interactions between the initiator(s) and target(s). We use `Agent` to denote the relation between an action and the action initiator, and use `Patient` to denote the relation between an action and the action target.

### 2.2.3 Causal And-Or graph

A causal And-Or graph (C-AOG) [10] captures the knowledge of the `Causal` relation between events and fluent changes. Figure 8(a) shows an example C-AOG. An And-node in a C-AOG represents a composition of conditions and events that can cause a fluent change. For example, to cause a door to open, it can be the case that the door is unlocked and someone pushes the door. An Or-node in a C-AOG represents alternative causes that can result in a fluent change. For example, the cause that a door is opened can be either that “someone unlocks and pushes the door” or that “the door is automatic and someone walks close”. A leaf node in a C-AOG is either an event or a fluent.

### 2.2.4 Joint S/T/C-AOG

Although S-AOG, T-AOG and C-AOG have been proposed and investigated separately before, in this work we show for the first time that the three types of AOGs are interrelated and form a joint S/T/C-AOG, as shown in Figure 9. An action represented by a leaf node in the T-AOG is connected via the `Agent`, `Patient` and `Location` relations to the action initiator, target and

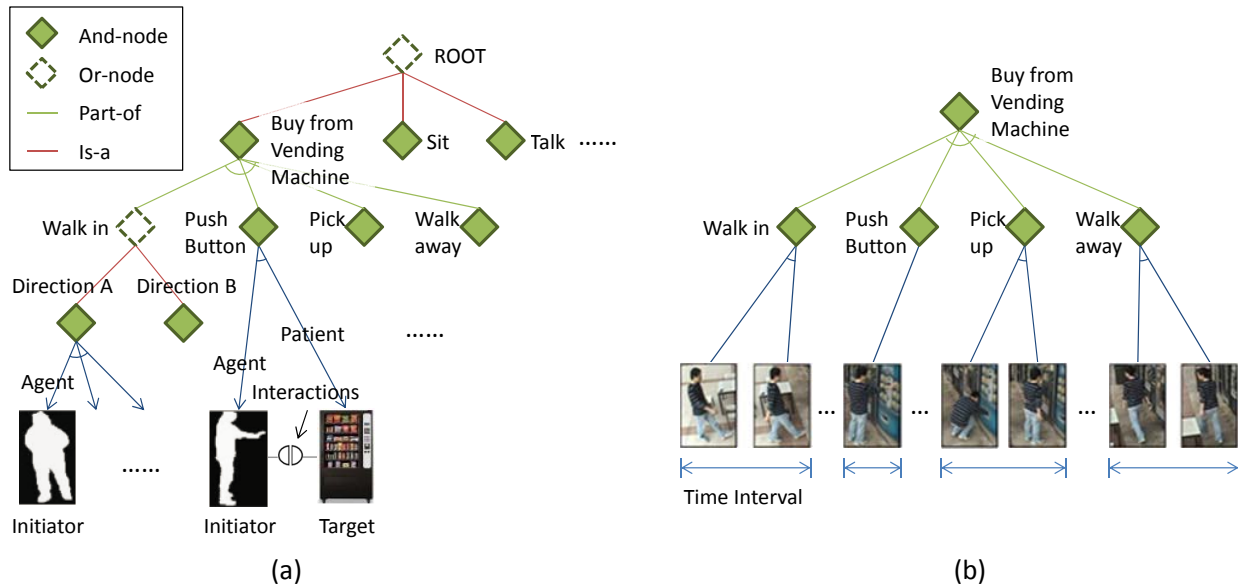


Fig. 7. (a) An example temporal And-Or graph (T-AOG). Each node represents a type of event or action; an And-node represents the temporal composition of its child nodes; an Or-node represents a selection among its child nodes; a leaf node represents an atomic action defined by the properties of and interactions between the action initiator(s) and target(s). (b) An example temporal parse graph, which is a realization of the T-AOG by making selections at Or-nodes.

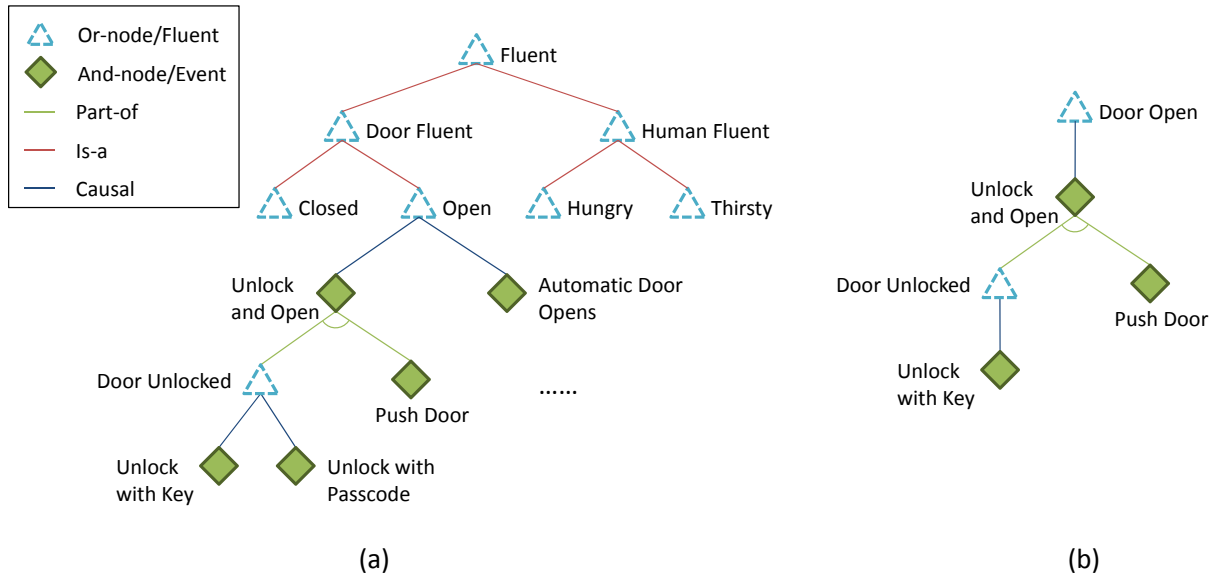


Fig. 8. (a) An example causal And-Or graph (C-AOG). An And-node represents the composition of its child nodes (conditions and sub-events); an Or-node represents a selection among its child nodes; a leaf node represents an event or a fluent. (b) An example causal parse graph, which is a realization of the C-AOG by making selections at Or-nodes.



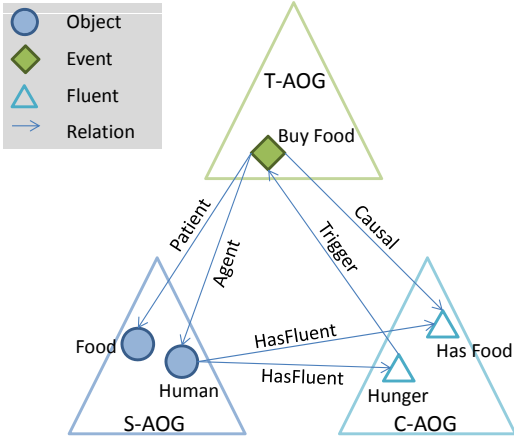


Fig. 9. An example of the relations between S-AOG, T-AOG and C-AOG.

location, which are objects modeled in the S-AOG. An event that has a *Causal* relation with a fluent change in the C-AOG is itself modeled by the T-AOG. There may also be a *Trigger* relation between a fluent in the C-AOG and an event in the T-AOG. Finally, there are *HasFluent* relations between objects in the S-AOG and fluents in the C-AOG. The connections across the three types of AOGs enable the propagation of information between them during parsing, which improves parsing accuracy. Connecting the three types of AOGs also leads to a more coherent and comprehensive joint interpretation, which facilitates subsequent applications such as query answering of who, what, when, where and why.

### 2.3 Parse graphs

We represent an interpretation of the input video and/or text with a graphical representation called a parse graph. A parse graph is a realization of the S/T/C-AOG by selecting at each Or-node one of the alternative configurations. For example, Figure 6(b), 7(b) and 8(b) show respectively a valid spatial, temporal and causal parse graph as realizations of the S/T/C-AOGs given in Figure 6(a), 7(a) and 8(a). A parse graph can be represented as a labeled directed graph. Each node in a parse graph represents an entity whose label indicates the semantic type of the entity. A node may also have a set of spatial-temporal annotations indicating the spatial-temporal coordinate or range of the entity represented by the node. A parse graph may also contain a special type of nodes representing attribute values in the form of strings or numbers. A directed edge between two nodes in a parse graph represents a relation between the two entities denoted by the two nodes. The label on the edge indicates the type of the relation. We adopt the *open world assumption* [50] which states that a parse graph may not be a complete interpretation of the scene and any statement not included or implied by the parse graph can be either true or false.

Figure 3(b) and 4(b) show the parse graphs that represent the interpretations of the videos and texts shown in Figure 3(a) and 4(a). Note that for better visualization of the parse graph, we use different node shapes to denote different types of entities, and use different edge colors to denote different types of relations; for nodes with temporal annotations (e.g., events), we lay them out in accordance with the time line; for a fluent that may have its value changed over time, we use levels to indicate the value of the fluent at each time point.

We infer parse graphs from video and text through maximizing the posterior probability that we shall discuss in the following few sections. As an explicit semantic representation of the video and text, our parse graphs facilitate subsequent applications such as text generation and query answering. In addition, our parse graph can be naturally represented as a Resource Description Framework (RDF) data model [51], one of the standard knowledge representations in the semantic web. This allows easy distribution of the parse graph data in the semantic web and enables the use of a multitude of approaches and tools for the editing, storage, inference and retrieval of the data.

### 2.4 Probabilistic Modeling

In this section we define a probabilistic distribution over parse graphs to account for their stochasticity. Since the valid parse graphs are specified by an AOG, we assign energy terms to the AOG to make it a probabilistic model of parse graphs (so an AOG embodies a stochastic grammar). First, at each Or-node of the AOG we specify an energy that indicates how likely each alternative configuration under the Or-node is selected in a parse graph. Second, at each And-node of the AOG we specify an energy for each relation between the child nodes of the And-node that captures the uncertainty of the relation. These energy terms can be either manually specified by domain experts or learned from data [52], [53], [10]. The energy of a parse graph is thus determined by the energy of configuration selections made at the Or-nodes and the energy of the relations between the child nodes of the And-nodes contained in the parse graph. Let  $V^{or}(pg)$  be the set of Or-nodes in  $pg$ ,  $E_{or}(v)$  the energy associated to the configuration selected at the Or-node  $v$ ,  $R(pg)$  the set of relations specified at the And-nodes in  $pg$ , and  $E_R(r)$  the energy associated with the relation  $r$ . The energy of a parse graph is defined as:

$$E(pg) = \sum_{v \in V^{or}(pg)} E_{or}(v) + \sum_{r \in R(pg)} E_R(r) \quad (2)$$

We also allow part of a parse graph to be missing with a penalty to accommodate incomplete observation or description.

As introduced in section 2.2, we use a joint S/T/C-AOG to define the valid parse graphs. The energy of a parse graph in our framework is therefore the summa-

tion of four energy terms.

$$E_{\text{STC}}(pg) = E_S(pg) + E_T(pg) + E_C(pg) + \sum_{r \in R^*(pg)} E_R(r) \quad (3)$$

where  $E_S(pg)$ ,  $E_T(pg)$  and  $E_C(pg)$  are the energy terms defined by the S-AOG, T-AOG and C-AOG respectively according to Eq.2,  $R^*(pg)$  is the set of relations that connect the three types of AOGs (as discussed in section 2.2.4) and are not contained in any of the AOGs, and  $E_R(r)$  is the energy associated with the relation  $r$ .

The prior probability of a parse graph  $pg$  is then defined as:

$$P(pg) = \frac{1}{Z} e^{-E_{\text{STC}}(pg)} \quad (4)$$

where  $Z$  is the normalization factor. Eq.4 defines one of the four factors in the posterior probability of the parse graphs given the input video and text (Eq.1). In the next three sections, we shall introduce and discuss the other three factors which are involved respectively in video parsing, text parsing and joint inference.

### 3 VIDEO PARSING

In video parsing, we aim to detect objects, scenes and events that form an interpretation of the input video. Since our prior knowledge of objects, scenes and events is encoded in the S/T/C-AOG, we regard the S/T/C-AOG as the prior of the video parse graph  $pg_{vid}$  and want to optimize the posterior of  $pg_{vid}$ . Our objective energy function for video parsing is:

$$E_v(pg_{vid}) = E_{\text{STC}}(pg_{vid}) - \log p(vid|pg_{vid}) \quad (5)$$

where  $E_{\text{STC}}$  is defined in Eq.3, and  $p(vid|pg_{vid})$  is the probability of the input video  $vid$  given the objects, scenes and events specified in the parse graph  $pg_{vid}$ . Since video parsing often contains a significant level of uncertainty, we output a set of candidate video parse graphs with low energy instead of a single best parse graph. To justify the use of the objective energy function  $E_v$  as defined in Eq.5, note that if we assume that the set of nodes and edges in the joint parse graph  $pg_{jnt}$  is a superset of that in the video parse graph  $pg_{vid}$ , then it can be shown that  $\exp(-E_v(pg_{vid}))$  is a factor of the joint posterior probability  $P(pg_{jnt}, pg_{vid}, pg_{txt}|vid, txt)$  (Eq.1); therefore, a video parse graph that has a high energy according to Eq.5 would very likely lead to a low joint posterior probability, so we use Eq.5 to prune such video parse graphs.

#### 3.1 Spatial parsing

We first perform spatial parsing on each video frame following the approach proposed in [7] called hierarchical cluster sampling. The approach consists of two stages. In the first stage, the approach performs bottom-up clustering in which lower-level visual entities (e.g., line segments) of a video frame are composed into possible higher level objects according to the S-AOG. To keep the

number of candidate objects tractable, compositions with high energy are pruned. In the second stage, the parser applies the Metropolis-Hastings algorithm in the space of spatial parse graphs. The sampler performs two types of operations to change the current spatial parse graph: the first is to add a candidate object composed in the first stage into the parse graph, where the proposal probability is defined based on the energy of the candidate object as well as the compatibility of the object with the rest of the parse graph; the second type of operation is to randomly prune the current parse graph. At convergence the approach outputs the optimal spatial parse graph.

#### 3.2 Temporal parsing

We perform temporal parsing following the approach proposed in [9], which is based on the Earley parser [54]. The input video is divided into a sequence of frames. The agents, objects and fluents in each frame are identified using the spatial parser and special detectors. The temporal parser reads in frames sequentially, and at each frame maintains a so-called state set that contains pending derivations of And-nodes that are consistent with the input video up to the current frame (i.e., for each And-node in the state set, only a subset of its child nodes have been detected from the video up to the current frame). At the beginning of parsing, the And-nodes at the top level of the T-AOG are added into the state set of frame 0, with all their child nodes pending. With each new frame read in, three basic operations are iteratively executed on the state set of the new frame. The *prediction* operation adds into the state set new And-nodes that are expected according to the existing pending derivations in the state set. The *Scanning* operation checks the detected agents, objects and fluents in the new frame and advances the pending derivations in the state set accordingly. The *Completion* operation identifies And-nodes whose derivations are complete and advances the pending derivations of their parent And-nodes. During this process, we prune derivations that have high energies to make the sizes of state sets tractable. After all the frames are processed, a set of candidate parses of the whole video can be constructed from the state sets.

#### 3.3 Causal parsing

After all the events are detected in temporal parsing, we then perform causal parsing. For each fluent change detected in the video using special detectors, we collect the set of events that occur within a temporal window right before the fluent change and run the Earley parser again based on the sub-graph of the C-AOG rooted at the detected fluent change.

After spatial, temporal and causal parsing is done, we combine the resulting spatial, temporal and causal parse graphs into a complete video parse graph. Figure 10 shows a parse graph of the example input video shown in Figure 4(a).

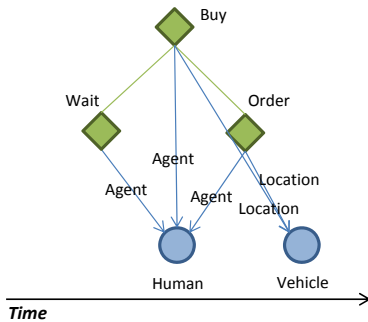


Fig. 10. A candidate parse graph of the video in our example input shown in Figure 4(a).

## 4 TEXT PARSING

In parallel to video parsing, we perform text semantic parsing to convert the input text description into the text parse graph  $pg_{txt}$ . Our semantic parsing approach is similar to many existing information extraction approaches, but our objective is distinctive in two aspects. First, we focus on the domain of narrative event descriptions to extract information about objects, events, fluent-changes and the relations between these entities. Important relations in this domain include spatial, temporal, functional and causal relations. Second, information extracted from the text is expressed using the same set of semantic types and parse graph representation that are also used in video parsing. This allows us to connect the two parse graphs and perform joint inference. We assume that text descriptions are unambiguous and therefore only a single text parse graph is produced from the input text. However, our approach can also handle text ambiguity and produce multiple candidate text parse graphs with different probabilities. Our semantic parsing of text is a processing pipeline consisting of four main steps: text filtering, part-of-speech tagging and dependency inference, dependencies filtering, and parse graph generation, which will be introduced in the following four subsections. In the case where the text descriptions are accompanied with time stamps or durations, we add temporal annotations to the corresponding event nodes in the resulting parse graph. We will use the parsing process of the sentence “Tom is buying food” as a running example in the rest of this section.

### 4.1 Text Filtering

This is a preprocessing step that locates and labels certain categories of words or phrases in the input text to facilitate subsequent text parsing steps. It first performs the named entity recognition (NER) to identify text elements related to names of persons, time, organizations and locations. Existing NER tools such as Stanford NER [55] can be used. Second, it recognizes certain compound noun phrases that refer to a single entity (e.g., “food truck” and “vending machine”). The words in a compound noun phrase are then grouped together into a

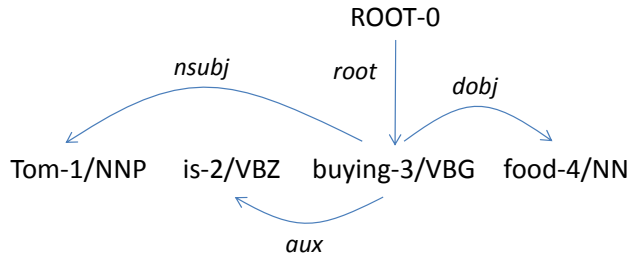


Fig. 11. An example dependency tree from step 2 of text parsing.

single word. Currently this is done by utilizing the chunker tool in Apache OpenNLP [56]. Our example sentence is not changed in this step.

### 4.2 Part-of-Speech Tagging and Dependency Inference

This step performs part-of-speech (POS) tagging and syntactic dependency parsing using the Stanford Lexicalized Parser [11]. During POS tagging, each word is assigned a grammatical category such as noun, verb, adjectives, adverb, article, conjunct and pronoun. The tagger uses the Penn treebank tag set which has 45 tags. The parser then performs dependency analysis to identify relations between the words in the sentence. It generates a dependency graph in which each node is a word and each directed edge represents a basic grammatical relation between the words. The Stanford Dependencies [57] include more than 50 types of grammatical relations such as subject, modifier, complement, direct and direct object. For example, the following dependency identifies Tom as the subject of the buying event.

$nsubj(buying, Tom)$ .

Together with POS labels, these dependency relations allow us to extract information from the text more easily.

For our example sentence, the result of this step is shown in Figure 11.

### 4.3 Dependencies Filtering

In this step, we map each word in the sentence to an entity type defined in the ontology or to a literal type. Examples of such mappings are shown in Table 1. This step is currently implemented using a look-up table but can also be automated by utilizing the WordNet dictionary [44].

In some cases, it is important to check the POS tag of a word to determine the correct entity type. For example, in the following two sentences:

The door is open(/JJ).  
They open(/VBP) the door.

the word “open” in the first sentence refers to a fluent state, while in the second sentence it refers to an action. The POS labels “JJ” and “VBP” allow us to map the

Entity type or literal type	Words in the input text
Human	person   individual   somebody   ...
Beverage	drink   beverage   ...
Buying	buying   purchasing   ...
Vending_machine	vending_machine   slot_machine   coin_machine   ...
Number	one   two   three   ...
Color	red   blue   ...
Aux	is   are

TABLE 1

Example mappings between words and entity types or literal types.

words to the correct entity types. Note that since we focus on a small set of entity types as specified in our ontology, words that can denote multiple entity types under the same POS tags are rare.

For our example sentence, the resulting dependency tree of this step is:

```
nsubj(Buying, Human_name)
aux(Buying, Aux)
root(ROOT-0, Buying)
dobj(Buying, Food)
```

We further replace each entity type with its top level supertype specified in the ontology. In our example, “Buying” is replaced with “Event” and “Food” is replaced with “Object”.

Compared with the original dependency tree which describes the relations between words, we have now added the entity types of these words. This provides additional cues to infer more precise semantic relations between the entities denoted by the words. For instance, from the above example, the dependency “nsubj(Event, Human\_name)” implies that the person (Tom) is the agent of the event (buying). Without the entity types, the dependency relation “nsubj” could be ambiguous, e.g., “nsubj(teacher, Tom)” from the sentence “Tom is a teacher” refers to an occupation attribute and not an agent-event relation.

#### 4.4 Parse Graph Generation

In order to extract semantic relations from dependencies more accurately, we need to consider not only individual dependencies but also their contexts (the semantic types of the words and the neighboring dependencies). Therefore, we design an attribute grammar to parse the set of dependencies and infer the semantic relations. The final text parse graph is then generated from the semantic relations inferred in this step as well as the entity types produced from the previous step.

In the attribute grammar, the terminal symbols are the mapped dependencies from the previous step. A set of production rules describe how the terminal symbols can

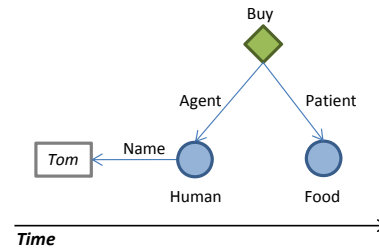


Fig. 13. The parse graph of the example input text “Tom is buying food”.

be generated from ROOT, the start symbol. The Earley parsing algorithm [54], [58] is used to parse the dependencies using the attribute grammar, which employs top-down dynamic programming to search for a valid set of production rules that generate the dependencies. The parse tree from the attribute grammar for our example sentence is shown in Figure 12.

In each node of the parse tree, information from the input sentence is organized into an attribute-value structure, which may include attributes such as the event or action type, person’s name, etc. Such information structures are propagated from the lower-level nodes to the higher-level nodes of the parse tree via the attribute functions associated with the production rules of the attribute grammar. For example, in the following production rule (which is used in the left-most branch of the parse tree in Figure 12):

```
nsubj(Event1, Object)
→ nsubj(Event2, Human_name)
```

the associated attribute functions are:

```
Event1.* := Event2.*
Object.* := Human_name.*
Object.type := Human
```

where “\*” is a wild-card character that stands for any subtext of the attribute name. After all the information from the input sentence is propagated to the top-level nodes in the parse tree of the attribute grammar, the final text parse graph is then generated based on these top-level nodes. For example, the top-level node Agent (Event, Object) in Figure 12 contains the following information:

```
Event.id = “buying-3”
Event.type = Buy
Object.id = “Tom-1”
Object.type = Human
Object.name = “Tom”
```

from which we can construct a part of the text parse graph containing the Agent relation between the buying event and the human as well as the name attribute of the human. The final text parse graph of the example sentence is shown in Figure 13.

In addition to extracting the Agent and Patient relations as shown in the running example, by using the attribute grammar we also extract the spatial, temporal

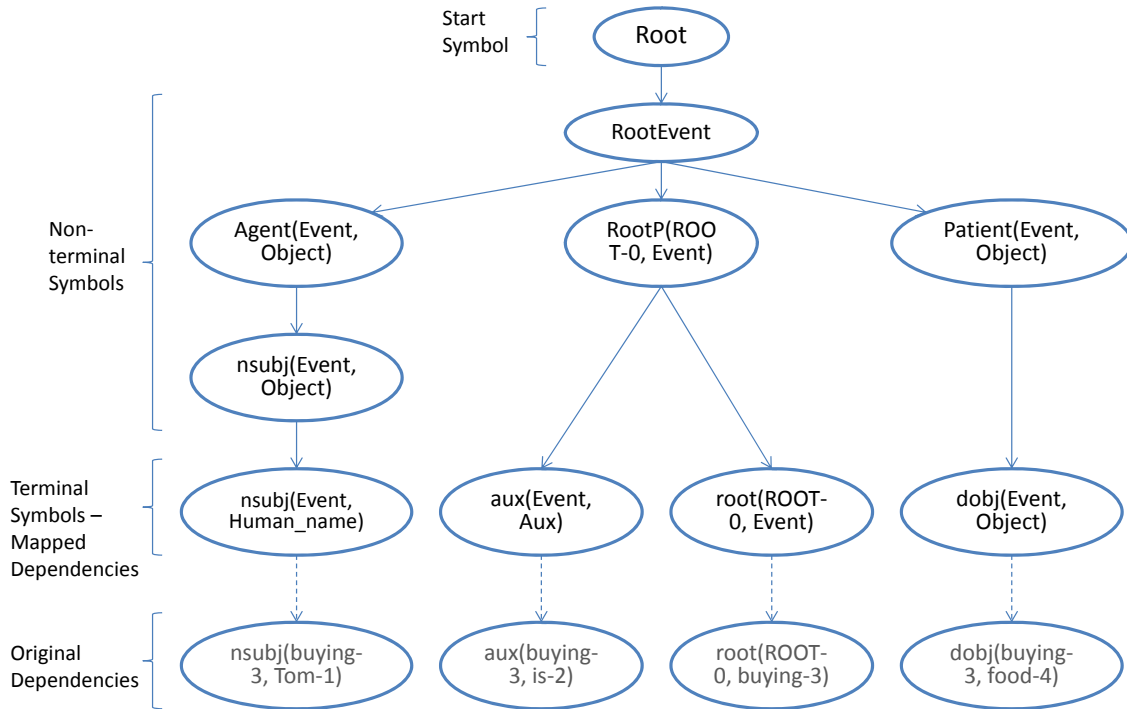


Fig. 12. An illustration of step 3 and 4 of text parsing: dependencies filtering of step 3 is shown in the bottom half of the figure; attribute grammar parsing of step 4 is shown in the top half of the figure.

Category	Relations	Function words
Spatial	location, locationFrom, locationTo	at, on, in, from, to, into
Temporal	timeAt, timeFrom, timeTo, timeBefore, timeAfter	at, from, to, before, after
Causal	causal, trigger	therefore, because
Functional	occupation, religion, race	is a

TABLE 2

Examples of relations extracted from text and the corresponding function words.

and causal relations as well as functional properties of objects based on the function words in the input sentence. Examples of such relations and the function words used to extract such relations are shown in Table 2. Some function words, such as “at”, “from” and “to”, have multiple meanings, but their semantic context in the input sentence allows the correct relations to be inferred during parsing.

## 5 JOINT INFERENCE

In joint inference, we construct the joint parse graph from the video and text parse graphs produced by the first two modules. Although spatial-temporal-causal parsing of videos and semantic parsing of text have been studied in the literature, this is the first time to integrate them together.

### 5.1 Challenges in Joint Inference

The joint parse graph shall not be a simple union of the video parse graph and the text parse graph. We need to consider the following three difficulties.

- 1) **Coreference.** The entities and relations covered by the two input parse graphs may be overlapping. Figure 14 shows such an example. We need to identify and merge the overlapping entities and relations. One challenge is that the same entity or relation might be annotated with different semantic types in the two parse graphs. For example, in Figure 14 the food truck entity is annotated with “Vehicle” in the video parse graph and “Food\_Truck” in the text parse graph. In addition, the spatial-temporal annotations of the same entity may not be exactly the same in the two parse graphs. As another challenge, for each entity there might be multiple candidate matching targets. For example, a person mentioned in the text may be matched to a few different persons in the video.
- 2) **Missing information.** Some information is not explicitly presented in the input video and text, but may be deduced using the prior knowledge encoded in the S/T/C-AOG. Such information is necessary to fill in the gap between the input video and text or is useful in downstream tasks like query answering. So we should include such information in the joint parse graph. Figure 15 shows an example in which the fluent “Has\_Food” is neither observable in the video nor stated in the

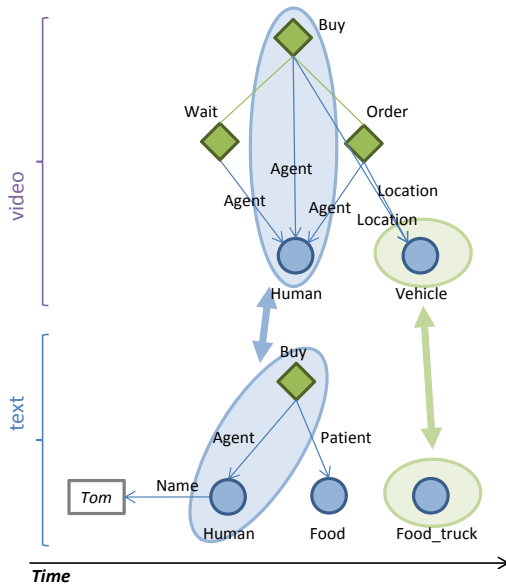


Fig. 14. An example of overlapping video and text parse graphs. The ovals and bidirectional arrows mark the sub-graphs that represent the same entities and relations.

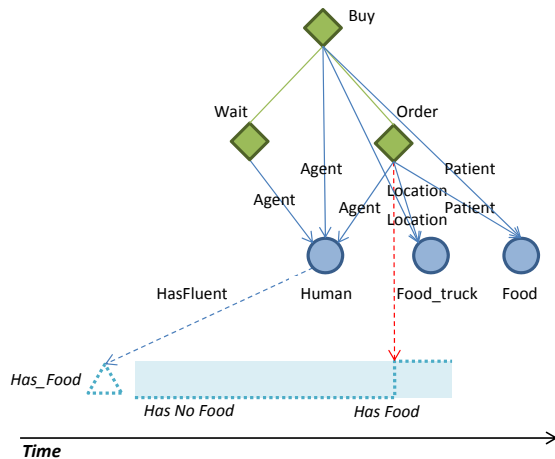


Fig. 15. An example of information that can be deduced from the parse graph. The dashed shapes and arrows represent inferred entities and relations.

text, but can be deduced from the event of buying food.

- 3) **Conflicts.** The parse graphs from the input video and text may contain conflicting information. Such conflicts manifest themselves once overlapping entities and relations from the two parse graphs are merged. Figure 16 shows an example parse graph that contains conflicting information. We need to detect such conflicts using the prior knowledge and correct them in the joint parse graph.

Figure 17 shows a schematic diagram of how the joint parse graph is constructed from the video and text parse graphs, which demonstrates the three difficulties in joint inference. There are two scenarios. In the first scenario

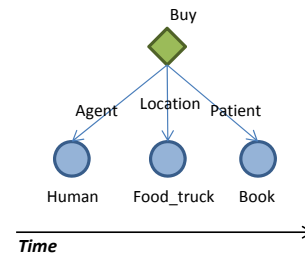
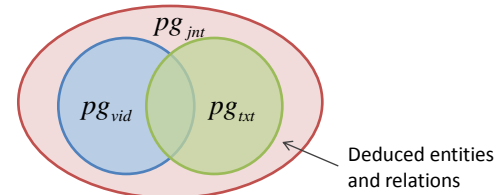
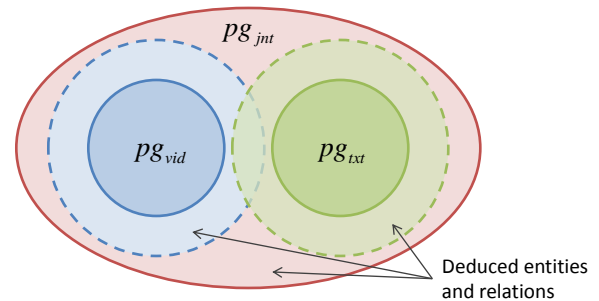


Fig. 16. An example of conflicting information in the parse graph. Here the conflict is between “buy book” and “buy from food truck”.



(a)



(b)

Fig. 17. A schematic diagram of how the joint parse graph  $pg_{jnt}$  is constructed from the video parse graph  $pg_{vid}$  and text parse graph  $pg_{txt}$ , when (a) there is overlapping between the video and text parse graphs, and (b) the video and text parse graphs have no overlapping but become overlapped after some steps of deduction.

(Figure 17(a)) where there is significant overlapping between the video and text parse graphs, we first need to identify the overlapping and solve the coreference problem, and then we can deduce missing information and correct conflicts to obtain a more accurate and comprehensive joint parse graph. In the second scenario (Figure 17(b)) where there is no overlapping between the video and text parse graphs, at first no coreference can be identified to connect the two parse graphs; only after the two parse graphs are expanded by deducing implicit information do they become overlapped, and then we can connect them by solving coreference and produce a coherent joint parse graph.

To solve the three difficulties, we propose three types of operators: *matching*, which matches and merges sub-graphs of the video and text parse graphs to solve the coreference problem; *deduction*, which inserts implicit

information to fill in the potential gap between video and text and makes the joint parse graph more comprehensive; and *revision*, which resolves possible conflicts between video and text. In the rest of this section, we first define the conditional probability  $P(pg_{vid}, pg_{txt} | pg_{jnt})$  in section 5.2, which specifies the relation between the joint parse graph and the video and text parse graphs; then we introduce the joint inference algorithm in section 5.3, which takes the union of the video and text parse graphs as the initial joint parse graph and iteratively changes it using the three types of operators to maximize the joint posterior probability.

## 5.2 Relating the three parse graphs

To relate the video parse graph  $pg_{vid}$ , the text parse graph  $pg_{txt}$  and the joint parse graph  $pg_{jnt}$ , we define the conditional probability  $P(pg_{vid}, pg_{txt} | pg_{jnt})$ . By multiplying it with the prior probability of the joint parse graph  $P(pg_{jnt})$  (based on Eq.4), we get the joint probability of the three parse graphs, which is used to guide our joint inference. Intuitively, the conditional probability  $P(pg_{vid}, pg_{txt} | pg_{jnt})$  models the generation of the video and text parse graphs given the joint parse graph, by considering how likely different types of entities and relations in the joint parse graph would be included in the video and text parse graphs, while penalizing the changes of the semantic, temporal and spatial annotations of these entities and relations in the video and text parse graphs. We do not assume independence of the generation of the video parse graph and that of the text parse graph, because in many scenarios they are indeed generated jointly, e.g., in TV news reporting the news editor would choose the footage and narrative that cover the same aspect of the whole news story. To make the maximum a posteriori estimation in joint inference tractable, we make the assumption that the inclusion and change of each entity or relation is independent of that of any other entity or relation in the parse graph, so we can factorize the conditional probability as defined below.

$$P(pg_{vid}, pg_{txt} | pg_{jnt}) = \prod_{x \in pg_{jnt}} \frac{1}{Z_x} e^{-E(x)} \quad (6)$$

$$\times \frac{1}{Z_\psi} e^{-\alpha_\psi \| (pg_{vid} \cup pg_{txt}) \setminus pg_{jnt} \|}$$

where  $x$  ranges over all the entities and relations in the joint parse graph  $pg_{jnt}$ ,  $Z_x$  and  $Z_\psi$  are normalization factors,  $\alpha_\psi$  is a constant representing the penalty of having an entity or relation in the video or text parse graph that is not contained in the joint parse graph, and  $E(x)$  is defined as follows.

$$E(x) = \begin{cases} \alpha_v(x) + d(x_j, x_v) & \text{if } x \in pg_{vid} \setminus pg_{txt} \\ \alpha_t(x) + d(x_j, x_t) & \text{if } x \in pg_{txt} \setminus pg_{vid} \\ \alpha_{vt}(x) + d(x_j, x_v) + d(x_j, x_t) & \text{if } x \in pg_{vid} \cap pg_{txt} \\ \alpha_\phi(x) & \text{if } x \notin pg_{vid} \cup pg_{txt} \end{cases}$$

We enforce the constraint that a relation in the joint parse graph can be contained in the video or text parse graph only if the two entities connected by the relation are also contained in the video or text parse graph. The notations used in the energy function are explained below.

- $\alpha_v(x)$  is the energy that  $x$  is contained in the video parse graph  $pg_{vid}$  but not in the text parse graph  $pg_{txt}$ . Ideally, this energy shall be set to a low value for elements that represent visible low-level details of objects, scenes and events, for example, the atomic actions of a drinking event (e.g., reaching to, lifting, holding and putting down a cup of water), which are seldom mentioned in the text description.
- $\alpha_t(x)$  is the energy that  $x$  is contained in the text parse graph  $pg_{txt}$  but not in the video parse graph  $pg_{vid}$ . Ideally, this energy shall be set to a low value for elements representing objects and events or their attributes that are either not visible in the video (e.g., a person's name, whether he is hungry) or hard to detect by the video parser (e.g., a person's gender can be hard to identify in a low resolution surveillance video).
- $\alpha_{vt}(x)$  is the energy that  $x$  is contained in both the video parse graph  $pg_{vid}$  and the text parse graph  $pg_{txt}$ . Ideally, this energy shall be set to a low value for elements representing visible high-level objects, scenes and events.
- $\alpha_\phi(x)$  is the energy that  $x$  is contained in neither the video parse graph  $pg_{vid}$  nor the text parse graph  $pg_{txt}$ . Ideally, this energy shall be set to a low value for elements representing low-level details of objects, scenes and events that are invisible or hard to detect in the video.
- $x_j$ ,  $x_v$  and  $x_t$  are the corresponding entities or relations of  $x$  in the joint parse graph  $pg_{jnt}$ , video parse graph  $pg_{vid}$  and text parse graph  $pg_{txt}$  respectively.
- $d$  is a distance measure that combines the semantic, temporal and spatial distances as defined below. It models the difference in the annotations of the entity or relation  $x$  in the joint parse graph and the video or text parse graph.

$$d(x, y) = \alpha_o d_o(x, y) + \alpha_t d_t(x, y) + \alpha_s d_s(x, y) \quad (7)$$

where  $\alpha_o$ ,  $\alpha_t$  and  $\alpha_s$  are constant weights, and the three distance measures  $d_o(x, y)$ ,  $d_t(x, y)$  and  $d_s(x, y)$  are explained below.

- The semantic distance  $d_o(x, y)$  is defined as the distance between the semantic types of  $x$  and  $y$  in the ontology. Several approaches, e.g., [59], [60], [61], can be used to measure the semantic distance given the taxonomy in an ontology. We require the distance to be very large if the two semantic types are disjoint (i.e., having no common instance).
- The temporal distance  $d_t(x, y)$  is defined as the time difference between the temporal annotations of  $x$  and  $y$  if both have temporal annotations.

- The spatial distance  $d_s(x, y)$  is defined as the Euclidean distance between the spatial annotations of  $x$  and  $y$  if both have spatial annotations.

Currently we set the values of all the constants in the definition heuristically, but it is possible to optimize them by learning from annotated training data (e.g., [62]).

### 5.3 Joint inference algorithm

We start with the simple union of the video and text parse graphs and then iteratively apply three types of operators that make changes to the joint parse graph.

- **Matching.** We match a node  $a$  from the video parse graph with a node  $b$  from the text parse graph, and merge them into a single node  $c$  which inherits all the edges attached to either node  $a$  or node  $b$ . To reduce the search space, we only match nodes that have a small distance between them (as defined in Eq.7). The semantic type of the new node  $c$  is set to be the more special one of the semantic types of node  $a$  and node  $b$ . For example, if node  $a$  is of semantic type “Vehicle” and node  $b$  is of semantic type “Food\_Truck”, then the semantic type of node  $c$  is set to “Food\_Truck” because “Food\_Truck” is a sub-type of “Vehicle”. The temporal and spatial annotations of nodes  $a$  and  $b$  are averaged before being assigned to the new node  $c$ . After the merging, if there exist two edges connecting to node  $c$  that have the same direction and a small distance (as defined in Equation 7), then the two edges are also merged.
- **Deduction.** We insert a new subgraph into the joint parse graph, such that the prior probability of the joint parse graph as defined by the S/T/C-AOG is increased. Specifically, for each entity in the joint parse graph, we find the type of this entity in the S/T/C-AOG and insert an instantiation of its immediate surrounding subgraph in the AOG into the joint parse graph. Because the energy defined in section 5.2 actually penalizes the size of the joint parse graph, we require the insertion to be minimal, i.e., each inserted node or edge does not duplicate any existing node or edge. This can be achieved by following the deduction operator with a series of matching operators to match and merge the inserted nodes and edges with the existing parse graph.
- **Revision.** We either remove a subgraph from the parse graph, or change the annotation of a node or edge in the parse graph, in order to solve a conflict as defined in the S/T/C-AOG (e.g., an instantiation of zero probability, or instantiation of more than one child of an Or-node). Because the energy defined in section 5.2 penalizes such revisions, we require the removal or change to be minimal, i.e., no extra node or edge is removed or changed that is unnecessary to solve the conflict.

Our objective function is the joint posterior probability defined in Eq.1. Because both the prior probability  $P(pg_{jnt})$  (Eq.4) and the conditional probability  $P(pg_{vid}, pg_{txt} | pg_{jnt})$  (Eq.6) can be factorized by the entities and relations in the parse graph, we can compute the change to the joint posterior probability caused by each operator.

For the surveillance data that we focus on in this paper, a typical parse graph contains a few tens of nodes and edges, so we use depth-first search with pruning to optimize the joint parse graph. For data of larger scales, it is straightforward to extend our algorithm with beam search. Given the initial joint parse graph made of a simple union of the video and text parse graphs, we first apply the matching operators until no further matching is possible, then apply the revision operators to solve conflicts, and finally apply the deduction operators (with the follow-up matching operators) to deduce implicit information. Each operator typically contains a number of possibilities (e.g., multiple nodes in the text parse graph can be matched with a specific node in the video parse graph), and we exam all the possibilities in a depth-first manner. In order to reduce the search space, we prune the possibilities that have significantly higher energy than the others.

**Stop criterion for deduction.** When applying the deduction operator, in some cases we may deduce multiple candidate subgraphs that have similar energy and are mutually exclusive (e.g., different children of an Or-node). In other words, there is significant uncertainty in the deduction as measured by information entropy. For example, we may observe a human in the video but cannot see what he is doing because of low resolution or occlusion, and based on the background knowledge we may infer a number of actions of the human that are equally possible. Since we adopt the open world assumption, it is reasonable to keep agnostic here and not add any new information. Specifically, we cancel the deduction operator if the entropy of the deduced subgraph is higher than a threshold:

$$\begin{aligned} H(pg_{de} | pg_{jnt}) &= - \sum_{i=1}^N P(pg_{de}^i | pg_{jnt}) \log P(pg_{de}^i | pg_{jnt}) \\ &> \frac{\log N}{c} \end{aligned}$$

where  $pg_{jnt}$  is the joint parse graph before applying a deduction operator,  $pg_{de}$  denotes the subgraph inserted by the deduction operator,  $N$  is the number of possible candidate subgraphs that can be deduced by the operator, and  $c > 1$  is a pre-specified constant.

Note that in video parsing we produce multiple candidate video parse graphs. So for each video parse graph we run the joint inference algorithm to find a joint parse graph, and then we output the parse graph triple  $\langle pg_{vid}, pg_{txt}, pg_{jnt} \rangle$  with the highest joint posterior probability. Although the algorithm described here only outputs a single optimal joint parse graph for each pair



of input video and text parse graphs, it is easy to adapt the algorithm to output multiple candidate joint parse graphs along with their joint posterior probabilities.

## 6 ANSWERING NATURAL LANGUAGE QUERIES

The joint parse graph is a semantic representation of the objects, scenes and events contained in the input video and text, which is very useful in many applications. In this section, we show how the joint parse graph can be used in semantic queries to answer questions, such as (a) questions in the forms of who, what, when, where and why; and (b) summary of scenes or events, e.g. how many persons.

### 6.1 Text query parsing

Given a simple plain text question about the objects, scenes and events in the input video and text, we parse the question into a formal semantic query representation. Since our joint parse graph can be represented in RDF, we represent a query using SPARQL, the standard query language for RDF [63].

The steps of text query parsing are identical to those in the text parsing module introduced in section 4. The attribute grammar for analyzing the dependencies is extended to include interrogative wh-words such as *who*, *what*, *when*, *where* and *why*. These wh-words indicate what the objective of the query is, and the rest of the text query defines the query conditions. For example, the question “Who has a cap?” is parsed into the following dependencies and attributes:

```
nsubj(Event, who)
root(ROOT-0, Event); Event.type = Possess
det(Object, Det)
doj(Event, Object); Object.type = Cap
```

The first dependency indicates that the query is asking for the agent of an event. The rest of the dependencies specify what the event type (“Possess”) and the event patient (“Cap”) are. These query conditions are then converted into the SPARQL format:

```
SELECT ?name ?who1
WHERE {
  ?who1 hasName ?name.
  ?cap4 rdf:type Cap.
  ?has2 rdf:type Possess.
  ?has2 hasAgent ?who1.
  ?has2 hasPatient ?cap4.
}
```

Queries that require aggregates can also be expressed in SPARQL via aggregate functions (COUNT, MAX, etc.) and grouping. For instance, the question “how many persons are there in the scene” is parsed into the following SPARQL query:

```
SELECT (COUNT(DISTINCT ?agent) as ?count)
WHERE {
```

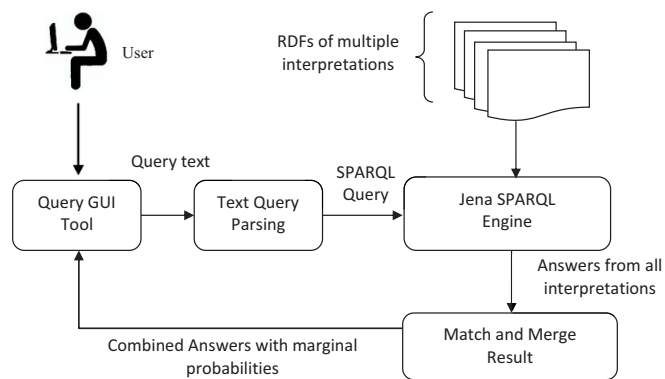


Fig. 18. The schematic diagram of querying multiple interpretations.

```
?agent rdf:type Human
}
```

After the SPARQL query is generated, it is evaluated by a SPARQL query engine which essentially performs graph matching of the patterns in the query condition with the joint parse graph stored in the RDF knowledge base. One implementation of the SPARQL query engine is included in Apache Jena [64]. The values obtained from the query engine are then processed to produce the answers to the query.

### 6.2 Computing answer probabilities from multiple interpretations

Our system can produce multiple joint parse graphs, each associated with its posterior probability. These joint parse graphs correspond to different interpretations of the input video and text. To answer a query accurately, we can execute the query on all the interpretations. The collected answers are then compared. We associate the answers from different interpretations by matching their semantic types and spatial-temporal annotations. For answers that are matched, their probabilities are combined. Formally, the probability of an answer  $a$  is computed as:

$$P(a) = \sum_{pg} P(pg) \mathbb{1}(pg \models a)$$

where  $P(pg)$  denotes the posterior probability of the joint parse graph  $pg$  and  $\mathbb{1}(pg \models a)$  is the indicator function of whether parse graph  $pg$  entails the answer  $a$ . In this way, different possible answers can be ranked based on their marginal probabilities. A schematic diagram of the process is shown in Figure 18.

### 6.3 Text query GUI tool

We developed a GUI tool for text query answering. A screenshot of this tool is shown in Figure 19. With this tool, the user types a query in natural language text and the system parses the query and retrieves the result,

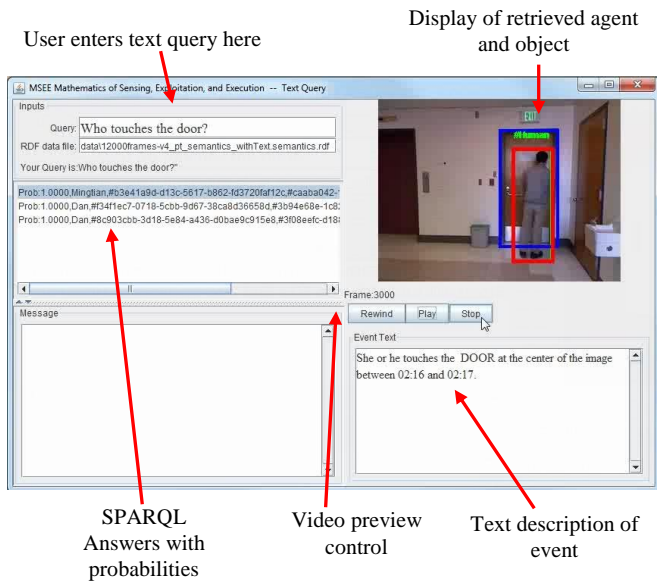


Fig. 19. Text Query GUI. See <http://www.stat.ucla.edu/~tukw/JointParsing/demo.html#demo2> for a demonstration video.

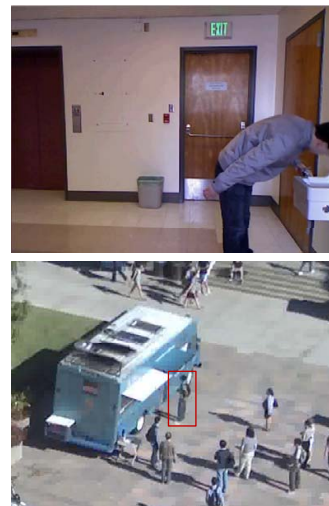
which is shown on the left of the GUI. If the answer is an event, the text description of the event is also presented in the bottom right of the GUI (the text descriptions are generated automatically from the joint parse graphs, which will be discussed in section 7.2.1). The corresponding video segment of the retrieved answer is shown on the top-right with overlays surrounding the agent and patient of the event. With this tool, the user can efficiently query about the events occurring in the video and review the relevant video segments. In our experiment, each query response typically took a few seconds.

## 7 EXPERIMENTS

### 7.1 Datasets

We collected two datasets of surveillance videos and text descriptions in indoor (hallway) and outdoor (courtyard) scenes. Compared with other types of datasets (such as movies with screenplays and news report footages with closed captions), surveillance videos contain a well-defined set of objects, scenes and events, which is amenable to controlled experiments; the text accompanying surveillance videos provides descriptions of the scenes shown in the video, while other types of datasets may contain text that are not descriptive of the scenes such as dialogs and comments; in addition, on surveillance videos we can control the level of details in the text descriptions and study its impact on the performance of joint parsing.

For each of the two datasets, we first recorded the video of the scene and divided the video into fifteen clips. Each video clip contains one or more events and lasts from a few seconds to over one minute. Based



(1) The person drinks water.  
 (2) The person walks to a water fountain. He bends down to drink water and then leaves.

(3) The person is thirsty.

(1) The person buys food at the food truck.

(2) The person walks to the food truck. He orders his food and waits. He picks up his food and walks away.

(3) The person is hungry.

Fig. 20. Examples from the hallway (top) and courtyard (bottom) datasets. The text descriptions shown on the right side are divided into three levels of details.

on the objects and events captured in the video, we manually constructed the ontology and the S/T/C-AOG for each dataset. We then asked five human subjects from different countries and background to provide text descriptions by watching the video clips. The human subjects were asked to only describe the events and the involved agents, objects and fluents in the scenes using plain English. The ontology was also provided to the human subjects to restrict the scope of their description. The human subjects were instructed to provide descriptions at three different levels of details, so we can study how the amount of text descriptions and their degrees of overlap with the video content can impact the performance of joint parsing. At the first level, the human subjects were told to give the simplest descriptions consisting of one or two sentences; at the second level, they were asked to give more details (typically less than ten sentences); at the third level, they were asked to provide information that is not directly observable in the video but can be inferred (typically zero to two sentences). Figure 20 shows two examples from our datasets.

### 7.2 Qualitative results

We ran our system to produce video parses, text parses and joint parses from the datasets. On a typical video-text pair, video parsing takes a few minutes to finish when running on a 12-node computer cluster, while text parsing and joint parsing finish within a few seconds on a desktop computer. We compared the information contained in the three types of parse graphs and made the following observations.

- Prominent objects and events are typically contained in both the video and the text parse graphs, e.g., a person drinking at the water fountain.

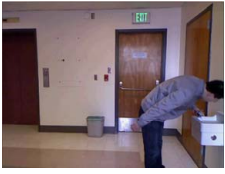

Input video	Input text	Generated text from the joint parse graph
	(Level 1) The person drinks water.	A person is thirsty. He approaches a fountain. He drinks water at a fountain. He leaves a fountain at the bottom of the image between 0:40 and 0:45.
	(Level 1-3) The person drinks water. He walks to a fountain. He bends down to drink water and then leaves. He is thirsty.	A person is thirsty. He approaches a fountain. He bends down. He drinks water at a fountain. He bends down because he wants to drink water. He leaves a fountain at the bottom of the image between 0:40 and 0:45.
	(Level 1) 0:00-1:02 The person waits for food.	A person is hungry. He waits for food between 0:00 and 1:02. He talks at a food truck between 0:00 and 0:38. He walks to a food truck between 0:39 and 0:44. He walks from a food truck between 0:52 and 1:02.
	(Level 1-3) 0:00-1:02 The person waits for food. 0:00-0:08 He stands and talks. 0:09-0:16 He bends down. 0:17-0:38 He squats. 0:39-0:45 He walks to the food truck. 0:46-0:51 He receives food. 0:51-0:54 He walks. 0:55-1:02 He walks away from the food truck. 0:00-1:02 He is hungry.	A person is hungry. He waits for food between 0:00 and 1:02. He stands between 0:00 and 0:08. He talks at a food truck between 0:00 and 0:08. He bends down between 0:09 and 0:16. He squats between 0:17 and 0:38. He walks to a food truck between 0:39 and 0:45. He receives food between 0:46 and 0:51. He walks away from a food truck between 0:51 and 1:02.

TABLE 3

Examples of generated text from the joint parse graphs. For each input video, we show the results produced from joint parsing using level 1 text and using text of all three levels.

- Low-level visible details of objects, events and fluents are often contained in the video parse graphs but not in the text parse graphs, e.g., a person approaching the water fountain before drinking and the door closing after a person enters.
- Two types of information are typically contained in the text parse graphs but not in the video parse graphs. The first is information invisible in the video but can be inferred, e.g., a person is thirsty (because he is drinking water). The second type of information is visible in the video but is not correctly detected by the video parser.
- The joint parse graphs may contain information that can be found in neither the video nor the text parse graphs, e.g., low-level details of objects and events that are not detected by the video parser, or invisible information that is not mentioned in the input text. Such information is deduced in joint inference from other information in the video and text parse graphs.

### 7.2.1 Text generation

We extend the text generator described in Yao et al. [65] to generate text descriptions from our joint parse graphs. The generated text presents the content of parse graphs in a user-friendly manner and helps non-expert users to understand the results from joint parsing.

The text generation process goes through two mod-

ules: sentence planner and sentence realizer. The sentence planner selects the content to be expressed, specifies hard sentence boundary, and provides information about the content. During this process, the information from the parse graph is converted to a functional description [66], which specifies elements of a sentence such as the event, subjects, objects, and their functional properties. With a functional description as the input, the sentence realizer generates the sentence by determining a grammatical form and performing word substitution. A head-driven phrase structure grammar (HPSG) [67] of English syntax is used to generate text sentences.

Table 3 shows some examples of the generated text. By comparing the input text and the generated text, it can be seen that when the input text is simple, the majority of the information in the joint parse graph comes from video parsing as well as deduction in joint inference; with more input text, an increasing part of the joint parse graph comes from text parsing.

We also evaluated the generated text based on the joint parse graph against the merged text descriptions from human subjects, using quantitative metrics such as BLEU [68] and METEOR [69] which were developed for evaluating machine translation. We find the resulting scores uninformative about the quality of joint parsing, because 1) the automatic text generator often chooses different words and expressions than human (e.g., “bend because he wants to drink” vs. “bend to drink”); 2)

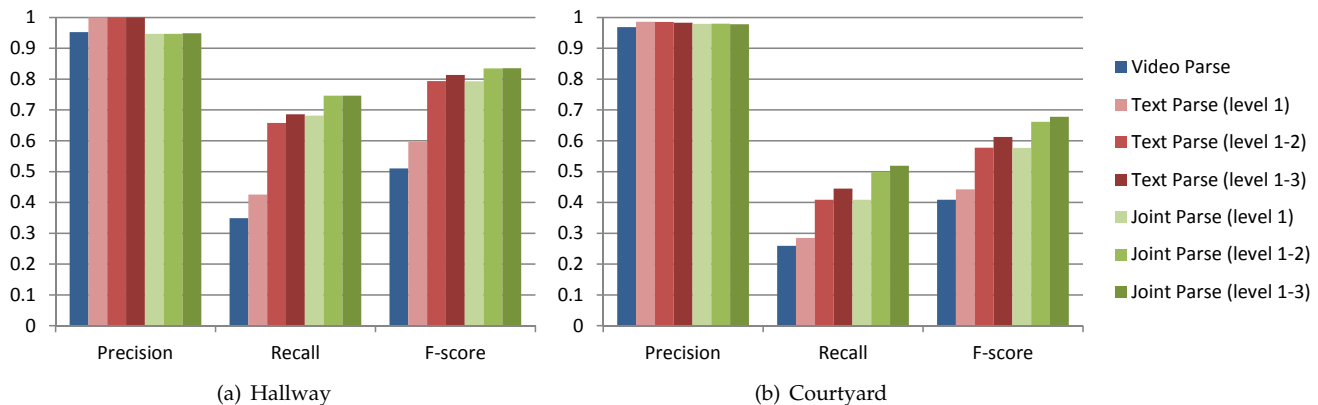


Fig. 21. The average precision, recall and F-score of the video, text and joint parse graphs in comparison against the ground truth parse graphs constructed from merged human text descriptions.

the generated text often contains information that is not included in the human text descriptions, such as the information from video parsing and deduction (e.g., low-level visual details). These differences are heavily penalized by the machine translation evaluation metrics, although they either are unimportant or should be rewarded for evaluating joint parse graphs. Therefore, we adopt two alternative quantitative evaluation approaches as introduced in the next subsection.

### 7.3 Quantitative evaluation

On each video clip in the datasets, for each human subject we used the text descriptions of the video clip as the text input to produce three joint parse graphs. The first joint parse graph is based on the level-1 text descriptions; the second is based on the combined text from level 1 and 2; and the third is based on the text of all three levels. We consider four baseline parse graphs: the video parse graph and the three text parse graphs based on level 1 text, level 1+2 text and text of all three levels. We evaluated the joint parse graphs as well as the baseline parse graphs using two different evaluation approaches.

#### 7.3.1 Comparison against ground truth

**Ground truth.** For each video clip, we constructed the ground truth parse graph by merging the parses of all the text descriptions from all the human subjects except the one whose text description was used as the input in the current run of experiments. The merged parse graph was manually checked to ensure correctness.

**Precision and recall.** We compared the parse graphs to be evaluated against the ground truth parse graph and measured the precision, recall and F-score. Let  $pg$  represent the parse graph to be evaluated,  $pg^*$  the ground truth parse graph,  $pg \cap pg^*$  the overlap between the two

parse graphs, and  $\|pg\|$  the size of parse graph  $pg$ .

$$\text{Precision} = \frac{\|pg \cap pg^*\|}{\|pg\|}$$

$$\text{Recall} = \frac{\|pg \cap pg^*\|}{\|pg^*\|}$$

F-score is defined as the harmonic mean of precision and recall. In computing precision and recall, in order to estimate the overlap between the parse graph to be evaluated and the ground truth parse graph, we ran depth-first search using the matching operator described in section 5.3 to match the nodes and edges in the two parse graphs. Since there might be information that is correct but missing from our ground truth parse graph constructed from merged human text descriptions, we also manually checked the parse graph to be evaluated when computing its precision.

Figure 21 shows the precision, recall and F-score of the baseline parse graphs and joint parse graphs averaged over all the video clips and human subjects of each dataset. It can be seen that all the precision scores are close to 1, suggesting the correctness of the information contained in all types of parse graphs. On the other hand, the recall and F-score of different types of parse graphs vary significantly. Each level of joint parse graphs has higher recall and hence higher F-score than the video parse graphs and the text parse graphs of the same level, which demonstrates the advantage of joint parsing. In particular, the joint parse graphs based on level-1 text have significantly higher recall and F-score than both the video parse graphs and the level-1 text parse graphs, implying that the information from the video and the level-1 text is highly complementary; it also shows that providing a simple text description (i.e., level-1 text) to a video can already boost the performance of video understanding to the level close to a full description of the video (i.e., all three levels of text combined). Adding more text (level 2 and 3) into joint parsing can be seen to further improve the joint parse graphs but with diminishing return.

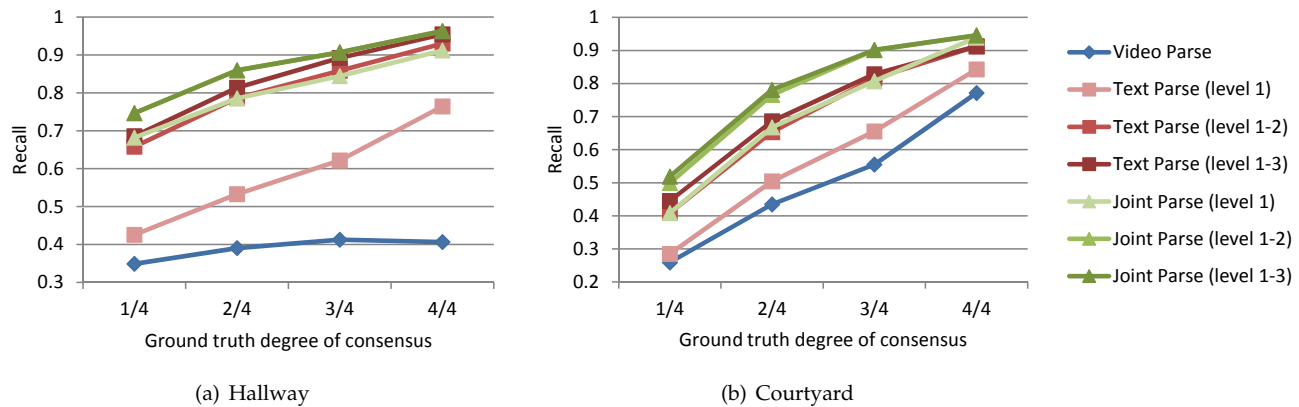


Fig. 22. The average recall of the video, text and joint parse graphs evaluated against the ground truth parse graphs re-constructed based on different degrees of consensus.

Comparing the results of the two datasets, we can see that the recall scores of the text and joint parse graphs on the courtyard dataset are much lower than those on the hallway dataset. This is because the courtyard dataset contains more complicated events and therefore the text descriptions from the five human subjects are often more different. Below is an example of such difference on a courtyard video clip:

- 1) The person walks.
- 2) The person walks *and talks*.
- 3) The person walks *out of a building*.
- 4) The person walks *across the courtyard*.

Since none of the text descriptions cover all the details of the event, the average recall of the text parse graphs is low. The joint parse graphs are produced with the text parse graphs as input, so their average recall is also lower on the courtyard dataset.

To compensate for this phenomenon in computing recall, we changed the ground truth parse graphs by keeping only the entities and relations that are mentioned by a minimal number of human subjects. The number of human subjects that mention an entity or relation indicates the degree of consensus among the human subjects regarding the existence and importance of the entity or relation. The maximal degree of consensus is 4/4 in our experiments because we used the text from four of the five human subjects to construct the ground truth parse graphs (excluding the provider of the input text of joint parsing). Figure 22 shows the average recall scores of all types of parse graphs evaluated against the re-constructed ground truth parse graphs based on different degrees of consensus. It can be seen that all the recall scores are improved with the increase of the ground truth degree of consensus. Most of the improvements are very large, implying that the text descriptions from different human subjects are indeed quite different, leading to significant changes of the ground truth parse graphs at different degrees of consensus. At the highest degree of consensus, the recall scores of most of the parse graphs are over 0.9, suggesting that these parse graphs

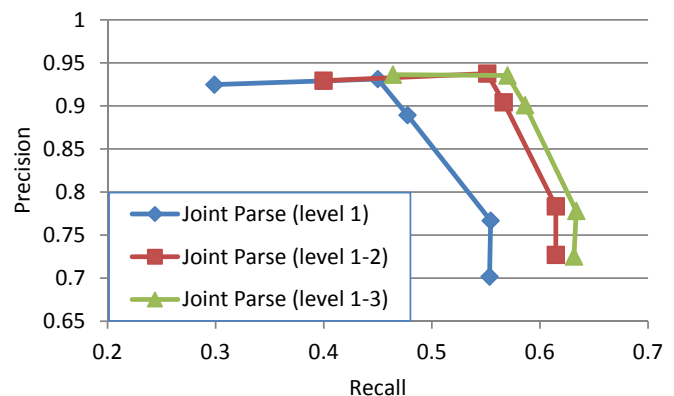


Fig. 23. The influence of the degree of deduction on the precision and recall of joint parse graphs.

cover the most important entities and relations in the scene. Among all types of parse graphs, only the video parse graphs on the hallway dataset do not have large improvement of recall, suggesting the limitation of the video parser on the hallway dataset.

**Degree of deduction.** We further studied the influence of the degree of deduction on joint parsing. On a subset of the courtyard dataset that contain complex events, we adjusted the value of  $\alpha_\phi$  to control the degree of deduction during joint inference and then measured the precision and recall of the resulting joint parse graphs. Figure 23 shows the results. It can be seen that with an increasing degree of deduction, at first the recall is greatly improved without decreasing the precision, suggesting that the deduced information is mostly correct; however, with further deduction the recall is improved modestly while the precision drops dramatically, which is caused by the increase of erroneous information being deduced; eventually, the recall stops improving while the precision continues to drop, implying that all the additional information being deduced is false.

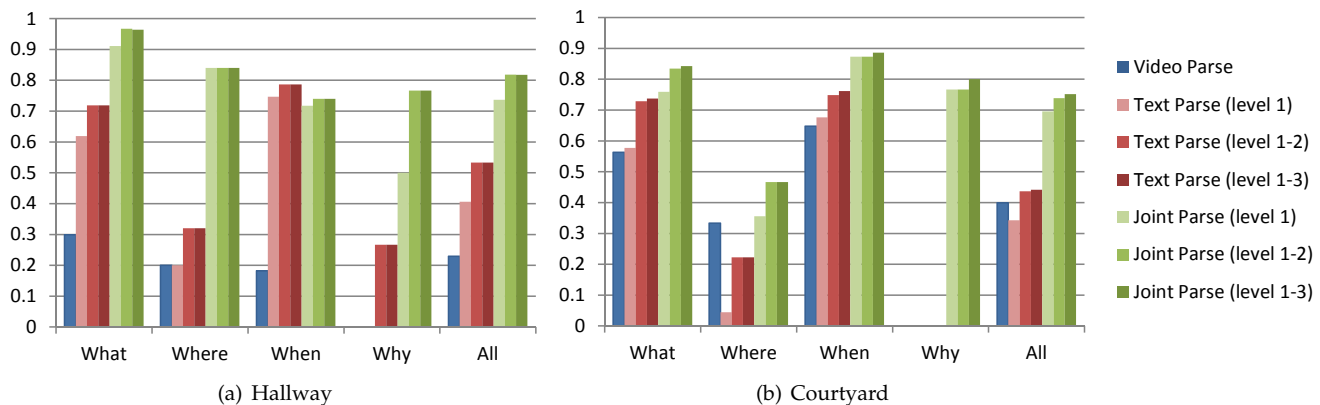


Fig. 24. The average F-score of the what, where, when and why queries and of all the queries based on the video, text and joint parse graphs in the query answering experiments.


Video clip	Queries	Correct answers
	Who is there in the designated area?	A human.
	What is he doing?	Buying food; waiting; walking; picking up food.
	Where does he buy food?	At the food truck.
	Why does he buy food?	He is hungry.
	When does he pick up food?	0:46-0:51.

TABLE 4

An example sequence of natural language queries and the correct answers.

### 7.3.2 Evaluation based on query answering

In section 6 we have introduced how the joint parse graph produced by our system can be used in query answering. Here we evaluate the quality of the joint parse graph by measuring the accuracy of query answering. This evaluation approach of video understanding is novel in that it directly measures the utility of the video understanding system in human-computer interaction, which goes beyond the conventional evaluation frameworks such as those based on classification or detection.

For each video clip in our datasets, we constructed a natural language query set that mimics the scenario in which a human user continuously queries the computer in order to acquire information of the objects, scene and events captured in the video clip. The natural language queries can take the forms of who, what, when, where and why. We also asked a human subject to provide the correct answers for each query set. Table 4 shows an example query set and the correct answers. We then entered the queries into our system and retrieved the answers. Please see <http://www.stat.ucla.edu/~tukw/JointParsing/demo.html#demo2> for a demonstration video of a user continuously asking queries through our GUI tool.

We compared the answers produced by the system against the correct answers provided by human. For each who/what/where/why query, we manually matched the system answers with the human answers and computed the precision (the percentage of the system answers that are correct) and recall (the percentage of the human answers that are returned by the system). For each when query, we checked the overlap between the time range returned by the system and that provided by human, based on which we computed the precision (the percentage of the time range returned by the system that is correct) and recall (the percentage of the correct time range that is covered by the system answer). We then computed the F-score which is the harmonic mean of precision and recall.

Figure 24 shows the average F-score of the what, where, when and why queries and of all the queries based on the video, text and joint parse graphs. The results of the who queries are not shown because the video clips used in the experiments either contain or designate a single person and therefore the who queries become trivial to answer. From the results we can see that query answering based on the joint parse graphs clearly outperforms that based on the video and text parse graphs. Note that in comparison with the experimental results from the first evaluation approach (Figure 21), in the query answering experiments the joint parse graphs have a significantly larger advantage over the video and text parse graphs. This is because answering a query involves perfect matching of the conditions of the query with one or more subgraphs of the parse graph, and therefore any small error in the parse graph would be magnified in the evaluation of query answering. Among the four types of queries shown in Figure 24, the joint parse graphs have the largest advantage on the why queries over the video and text parse graphs, because the causal relations required for answering the why queries are typically not detected in video parsing and not always mentioned in the input text but can be inferred by the deduction operator in

joint inference. The advantage of the joint parse graphs on the where queries is also large, because in video parsing many events being queried are not detected, and in the text the location information is sometimes skipped (as can be seen in the examples in Figure 20), while in joint parsing additional location information can be deduced. For the when queries, the average F-score of the text parse graphs is well below 1.0 although the time annotations provided by human subjects in the input text are quite accurate, which is because some of the events being queried are not described in the input text and therefore the corresponding when queries cannot be answered based on the text parse graphs.

## 8 CONCLUSION AND DISCUSSION

We propose a multimedia analysis framework for parsing video and text jointly for understanding events and answering queries. Our framework produces a parse graph of video and text that is based on the spatial-temporal-causal And-Or graph (S/T/C-AOG) and represents the compositional structures of spatial information (objects and scenes), temporal information (actions and events) and causal information (causalities between events and fluents). We present a probabilistic generative model for joint parsing that captures the relations between the input video/text, their corresponding parse graphs and the joint parse graph. Based on the model we propose a joint parsing system that consists of three modules: video parsing, text parsing and joint inference. In the joint inference module, we produce the joint parse graph by performing matching, deduction and revision on the video and text parse graphs. We show how the joint parse graph produced by our system can be used to answer natural language queries in the forms of who, what, when, where and why. The empirical evaluation of our system shows satisfactory results.

For future work, first, we will further improve both video parsing and text parsing with respect to parsing quality and efficiency. Second, currently we manually construct the S/T/C-AOG to model objects, scenes and events; in the future we want to investigate automatic approaches to learning AOG from data, which would facilitate the application of our approach to video-text data from novel domains. Third, we plan to test our system on video and text data more challenging than surveillance videos with human intelligence descriptions, e.g., news report data, which contains more diversified contents and larger gap between video and text and requires more sophisticated background knowledge in joint parsing.

## ACKNOWLEDGMENT

The work is supported by the DARPA grant FA 8650-11-1-7149, the ONR grant N00014-10-1-0933 and N00014-11-C-0308, and the NSF CDI grant CNS 1028381. We want to thank Mingtian Zhao, Yibiao Zhao, Ping Wei, Amy Morrow, Mohamed R. Amer, Dan Xie and Sinisa Todorovic for their help in automatic video parsing.

## REFERENCES

- [1] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *CVPR*, 2004.
- [2] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan, "Matching words and pictures," *J. Mach. Learn. Res.*, vol. 3, pp. 1107–1135, Mar. 2003.
- [3] F. Monay and D. Gatica-Perez, "Modeling semantic aspects for cross-media image indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1802–1817, 2007.
- [4] C. Wang, D. Blei, and F.-F. Li, "Simultaneous image classification and annotation," in *CVPR*, 2009.
- [5] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press, 1999.
- [6] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 4, pp. 259–362, 2006.
- [7] Y. Zhao and S. C. Zhu, "Image parsing with stochastic scene grammar," in *NIPS*, 2011.
- [8] Y. Zhao and S.-C. Zhu, "Scene parsing by integrating function, geometry and appearance models," in *CVPR*, 2013.
- [9] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing video events with goal inference and intent prediction." in *ICCV*, 2011.
- [10] A. Fire and S.-C. Zhu, "Using causal induction in humans to learn and infer causality from video," in *CogSci*, 2013.
- [11] "The stanford parser: A statistical parser," <http://nlp.stanford.edu/software/lex-parser.shtml>.
- [12] G. Iyengar, P. Duygulu, S. Feng, P. Ircing, S. P. Khudanpur, D. Klakow, M. R. Krause, R. Manmatha, H. J. Nock, D. Petkova, B. Pytlik, and P. Virga, "Joint visual-text modeling for automatic retrieval of multimedia documents," in *MULTIMEDIA*, 2005.
- [13] J. Yang, Y. Liu, E. P. Xing, and A. G. Hauptmann, "Harmonium models for semantic video representation and classification," in *SDM*, 2007.
- [14] C. Xu, Y. Zhang, G. Zhu, Y. Rui, H. Lu, and Q. Huang, "Using webcast text for semantic event detection in broadcast sports video," *IEEE Transactions on Multimedia*, vol. 10, no. 7, pp. 1342–1355, 2008.
- [15] S. Paek, C. L. Sable, V. Hatzivassiloglou, A. Jaimes, B. H. Schiffman, S.-F. Chang, and K. R. McKeown, "Integration of visual and text-based approaches for the content labeling and classification of photographs," in *ACM SIGIR*, 1999.
- [16] D. M. Blei, "Probabilistic models of text and images," Ph.D. dissertation, University of California at Berkeley, 2004.
- [17] T. L. Berg and D. A. Forsyth, "Animals on the web," in *CVPR*, 2006.
- [18] Y. Liu, D. Xu, I. W.-H. Tsang, and J. Luo, "Textual query of personal photos facilitated by large-scale web data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 1022–1036, 2011.
- [19] Y. Jia, M. Salzmann, and T. Darrell, "Learning cross-modality similarity for multinomial data," in *ICCV*, 2011.
- [20] Y. Feng and M. Lapata, "Automatic caption generation for news images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, 2013.
- [21] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 113–140, 2005.
- [22] Y. Jin and S. Geman, "Context and hierarchy in a probabilistic image model," in *CVPR*, 2006.
- [23] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *CVPR*, 2007.
- [24] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 59–73, Jan. 2009.
- [25] J. Porway, Q. Wang, and S. C. Zhu, "A hierarchical and contextual model for aerial image parsing," *International journal of computer vision*, vol. 88, no. 2, pp. 254–283, 2010.
- [26] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *CVPR*, 1997.
- [27] P. Natarajan and R. Nevatia, "Coupled hidden semi markov models for activity recognition," in *WMVC*, 2007.
- [28] S. Gorga and K. Otsuka, "Conversation scene analysis based on dynamic bayesian network and image-based gaze detection," in *ICMI-MLMI*, 2010.
- [29] M. S. Ryoo and J. K. Aggarwal, "Recognition of composite human activities through context-free grammar based representation," in *CVPR*, 2006.

- [30] S.-W. Joo and R. Chellappa, "Recognition of multi-object events using attribute grammars," in *ICIP*, 2006.
- [31] Z. Zhang, T. Tan, and K. Huang, "An extended grammar system for learning and recognizing complex visual events," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 240–255, Feb. 2011.
- [32] M. R. Amer, D. Xie, M. Zhao, S. Todorovic, and S.-C. Zhu, "Cost-sensitive top-down/bottom-up inference for multiscale activity recognition," in *ECCV*, 2012.
- [33] A. Hakeem, Y. Sheikh, and M. Shah, "CASE<sup>E</sup>: a hierarchical event representation for the analysis of videos," in *AAAI*, 2004.
- [34] W. Brendel, A. Fern, and S. Todorovic, "Probabilistic event logic for interval-based event recognition," in *CVPR*, 2011.
- [35] R. J. Mooney, "Learning for semantic parsing," in *CICLing*, 2007.
- [36] H. Poon and P. Domingos, "Unsupervised semantic parsing," in *EMNLP*, 2009.
- [37] P. Liang, M. I. Jordan, and D. Klein, "Learning dependency-based compositional semantics," in *ACL*, 2011.
- [38] J. R. Hobbs and R. Mulkar-mehta, "Using abduction for video-text coreference," in *Proceedings of BOEMIE 2008 Workshop on Ontology Evolution and Multimedia Information Extraction*, 2008.
- [39] C. Liu, R. Fang, and J. Y. Chai, "Towards mediating shared perceptual basis in situated dialogue," in *SIGDIAL*, 2012.
- [40] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, Jun. 1993.
- [41] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu, "The evolution of protégé: an environment for knowledge-based systems development," *Int. J. Hum.-Comput. Stud.*, vol. 58, no. 1, pp. 89–123, Jan. 2003.
- [42] O. Corcho, M. Fernández-López, A. d. M. G. Pérez, and O. Vicente, "Webode: An integrated workbench for ontology representation, reasoning, and exchange," in *EKAW*, 2002.
- [43] L. Zhang, Y. Yu, J. Lu, C. Lin, K. Tu, M. Guo, Z. Zhang, G. T. Xie, Z. Su, and Y. Pan, "Orient: Integrate ontology engineering into industry tooling environment," in *ISWC*, 2004.
- [44] "WordNet: A lexical database for English," <http://wordnet.princeton.edu/>.
- [45] D. Foxvog, "Cyc," in *Theory and Applications of Ontology: Computer Applications*. Springer, 2010, pp. 259–278.
- [46] A. Maedche and S. Staab, "Ontology learning for the semantic web," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 72–79, Mar. 2001.
- [47] P. Buitelaar and B. Magnini, "Ontology learning from text: An overview," in *In Ontology Learning from Text: Methods, Applications and Evaluation*, 2005.
- [48] P. Cimiano, *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [49] E. T. Mueller, *Commonsense Reasoning*. Morgan Kaufmann, 2006.
- [50] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, December 2002.
- [51] "RDF/XML syntax specification (revised)," <http://www.w3.org/TR/REC-rdf-syntax/>, 2004.
- [52] Z. Si and S. Zhu, "Learning and-or templates for object modeling and recognition," *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2012.
- [53] Z. Si, M. Pei, B. Yao, and S.-C. Zhu, "Unsupervised learning of event and-or grammar and semantics from video," in *ICCV*, 2011.
- [54] J. Earley, "An efficient context-free parsing algorithm," *Commun. ACM*, vol. 26, no. 1, pp. 57–61, Jan. 1983.
- [55] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *ACL*, 2005.
- [56] "Apache OpenNLP," <http://opennlp.apache.org/>.
- [57] M.-C. de Marneffe and C. D. Manning, "Stanford typed dependencies manual," 2008.
- [58] A. Hakeem, M. W. Lee, O. Javed, and N. Haering, "Semantic video search using natural language queries," in *MM*, 2009.
- [59] W.-N. Lee, N. Shah, K. Sundlass, and M. Musen, "Comparison of ontology-based semantic-similarity measures," *AMIA Annu Symp Proc.*, vol. 2008, pp. 384–388, 2008.
- [60] R. Thiagarajan, G. Manjunath, and M. Stumptner, "Computing semantic similarity using ontologies," HP Labs, Tech. Rep., 2008.
- [61] C. Pesquita, D. Faria, A. Falcao, P. Lord, and F. Couto, "Semantic similarity in biomedical ontologies," *PLoS Comput Biol.*, vol. 5, no. 7, 2009.
- [62] J. Dodge, A. Goyal, X. Han, A. Mensch, M. Mitchell, K. Stratos, K. Yamaguchi, Y. Choi, H. Daumé III, A. C. Berg, and T. L. Berg, "Detecting visual text," in *NAACL*, 2012.
- [63] "SPARQL query language for RDF," <http://www.w3.org/TR/rdf-sparql-query/>, 2008.
- [64] "Apache jena," <http://jena.apache.org/>.
- [65] B. Yao, X. Yang, L. Lin, M. W. Lee, and S. C. Zhu, "I2T: Image parsing to text description," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1485–1508, 2010.
- [66] I. Langkilde and K. Knight, "Generation that exploits corpus-based statistical knowledge," in *ACL-COLING*, 1998.
- [67] C. Pollard and I. A. Sag, *Head-driven phrase structure grammar*. University of Chicago Press, 1994.
- [68] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002.
- [69] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.