Recognizing Player Goals in Open-Ended Digital Games with Markov Logic Networks

Eun Y. Ha, Jonathan P. Rowe, Bradford W. Mott, and James C. Lester

Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
{eha, jprowe, bwmott, lester}@ncsu.edu

Abstract

Goal recognition in digital games centers on identifying the concrete objectives that a player is attempting to achieve given a domain model and a sequence of player actions in a virtual environment. Goal recognition models in open-ended digital games introduce opportunities for adapting gameplay events based on the choices of individual players, as well as interpreting player behaviors during post hoc data mining analyses. However, goal recognition in open-ended games poses significant computational challenges, including inherent uncertainty, exploratory actions and ill-defined goals. This chapter reports on an investigation of Markov logic networks (MLNs) for recognizing player goals in open-ended digital game environments with exploratory actions. The goal recognition model was trained on a corpus collected from player interactions with an open-ended game-based learning environment, CRYSTAL ISLAND. We present experimental results, in which the goal recognition model was compared to *n*-gram models. The findings suggest the proposed goal recognition model yields significant accuracy gains beyond the *n*-gram models for predicting player goals in an open-ended digital game.

Keywords: Goal recognition, digital games, educational games, Markov logic networks, *n*-grams, plan recognition.

1. Introduction

Goal recognition, and its more general form of plan recognition, are classic problems in artificial intelligence (Carberry, 2001; Kautz, 1986; Schmidt, Sridharan, and Goodson, 1978). Goal recognition is the process of identifying the high-level objective that an agent is trying to achieve based on an observed sequence of actions performed by the agent. It is a restricted form of plan recognition, a task that involves identifying both the agent's goal and the action sequence that will achieve that goal based on an observed sequence of actions. These tasks, plan recognition and goal recognition, are connected by their shared focus on recognizing patterns in action sequences, and they are widely regarded as key components of human intelligence. The capacity to interpret patterns in others' actions is critical for navigating everyday situations, such as driving a car, holding a conversation, or playing a sport. Consequently, computational models of plan recognition are relevant to a broad range of applications. Plan and goal recognition models have been devised for story understanding systems (Charniak and Goldman, 1993), productivity software (Horvitz, et al., 1998), intelligent tutoring systems (Conati, Gertner, and VanLehn, 2002), and dialogue systems (Blaylock and Allen, 2003). More recently, related models for activity recognition have been used to create assistive technologies for the disabled (Chu, Song, Levinson, and Kautz, 2012) as well as aircraft refueling recognizers that operate on video data (Hoogs and Perera, 2008). An emerging, promising application area for goal recognition systems is digital games (Gold, 2010; Kabanza, et al., 2010; Laviers and Sukthankar, 2011; Mott and Lester, 2006; Orkin, 2007). Digital games vary considerably in genre and gameplay style, giving rise to many formulations of goal recognition tasks.

Digital games continue to grow increasingly complex in their graphical presentations, interaction designs, and simulation capabilities. However, digital games rarely demonstrate capabilities for interpreting players' low-level actions at higher levels—in other words, recognizing patterns in player behavior—to dynamically adapt game events. A key example of this missing capability is games' inability to recognize players' goals. Recognizing a player's goals in a digital game involves identifying the concrete objectives that a player is attempting to achieve given a domain model and sequence of player actions in the virtual environment. In theory, games have full access to fine-grained information about players' actions, and this data could be utilized by goal recognition models to drive how games respond to players' behavior at run-time. Yet digital games have traditionally relied on comparatively unsophisticated methods such as scripts, triggers, and state machine-driven approaches for reacting to player behavior, leading to rigid and sometimes artificial results (Yannakakis, 2012).

If adopted, goal recognition models offer several prospective benefits to game creators. First, they expand the potential of *player-adaptive games*, which dynamically adapt game events in response to players' actions (Riedl, Saretto, and Young, 2003). As an illustrative example, consider a game in which a player forms a goal to give some food, an apple, to a hungry villager. Later in the narrative, unbeknownst to the player, the apple will emerge as an important element of the game's narrative; the player will discover that the apple is transmitting an infectious disease, which the player must diagnose. Giving the apple to the villager conflicts with this planned narrative. The player will not be able to later examine the apple if he has previously fed it to the hungry villager. Without a goal recognition model, the game is unable to interpret the player's actions as anything more than a sequence of movements, object manipulations, and conversations. A goal recognition model enables the player-adaptive game to recognize the player's goal, and subsequently infer that the goal conflicts with the planned narrative. Using knowledge about the player's goal, the player-adaptive game can dynamically augment the game experience, such as by directing the villager to refuse the apple.

Second, *game-based learning environments* stand to benefit from goal recognition models. Interpreting players' goals and plans are valuable in assessment, particularly for subjects where the player is expected to demonstrate a problem-solving strategy or complex multi-step skill in the virtual environment. More generally, goal recognition models can inform intelligent tutoring systems embedded within game-based learning environments (Johnson, 2010; Lee, Mott, and Lester, 2012). Accurately diagnosing students' problem-solving goals is essential for intelligent tutors to assess what concepts and skills a player understands, as well as possible misconceptions that the player may possess. These capabilities enable intelligent tutoring systems to provide hints or remediation tailored to individual learners.

Third, goal recognizers can provide valuable information for *telemetry efforts* by game developers. Game telemetry involves remotely collecting, storing, and analyzing log data about player behaviors, yielding information about how players use digital games "in the wild" (Zoeller, 2010). Over the past several years, telemetry has been the subject of growing interest as game developers seek information to inform future game designs, such as which items are most popular or which regions of virtual maps cause gameplay problems. Adding goal recognition models to developers' suite of game telemetry tools facilitates high-level interpretations of players' raw interaction data. Goal recognition models equip analysts with tools to identify players' intentions, determine which gameplay strategies are most successful, and adjust parts of the game that cause frustration or lead to players quitting.

While goal recognition models show promise for a broad range of games, their potential for enriching open-ended (or "sandbox") games is especially notable. Open-ended games, such as the popular Elder Scroll series (Bethesda, 2011) and Minecraft (Mojang, 2009), feature expansive environments and multiple paths for accomplishing in-game objectives (Squire, 2008). In these immersive 3D games players choose the goals they want to pursue, and they develop their own plans to achieve the goals. Goals can be formed in several ways in open-ended games: the software may explicitly present goals for the player to achieve, or the player may implicitly define the goals for herself during gameplay, effectively hiding the goals from the software. In either of these cases, goals may be well defined (e.g., Retrieve the magical staff from the dragon's cave and return it to the magician) or they may be ill defined (e.g., Build a castle on the mountain). Players may perform actions in deliberate sequences in order to accomplish goals, or they may perform actions in an exploratory manner, inadvertently encountering or achieving goals. Successive goals within a game may be independent of one another, or they may have complex inter-relationships defined by overarching narrative structures. Each of these variations introduce substantive challenges for goal recognition.

Goal recognition models in digital games must cope with several sources of inherent uncertainty. A single sequence of player actions is often explainable by multiple possible player goals, and a given goal can be achieved through a wide range of possible action sequences. While logical representations enable concise encodings of structural relations in goal recognition domains, they lack support for reasoning about uncertainty. Markov logic networks (MLNs) provide a formalism that unifies logical and probabilistic representations into a single framework, preserving the representational properties of first-order logic while supporting reasoning about uncertainty as made possible by probabilistic graphical models (Richardson and Domingos, 2006). This chapter reports on an investigation of Markov logic networks for recognizing player goals in an open-ended digital game environment with exploratory actions. Specifically, we evaluate the predictive accuracy of MLN-based goal recognition models for a story-centric game-based learning environment.

In this chapter, we survey prior work on player goal recognition in digital games. We also motivate our decision to use Markov logic networks for goal recognition, comparing our framework to other computational approaches for goal recognition in various domains, as well as recent work leveraging MLNs for related tasks. We describe a MLN goal recognition model that was trained on a corpus collected from player interactions with an open-ended game-based learning environment called CRYSTAL ISLAND. In CRYSTAL ISLAND, players are assigned a single high-level objective: solve a science mystery. Players interleave periods of exploration and deliberate problem solving in order to identify a spreading illness that is afflicting a research team stationed on the island. In this setting, goal recognition involves predicting the next narrative sub-goal that the player will complete as they solve the interactive mystery. We present findings that suggest the MLN goal recognition model yields significant accuracy gains beyond alternative probabilistic approaches for predicting player goals. We discuss the study's implications, both in terms of algorithmic techniques for goal recognition as well as methodologies that leverage digital games for investigating goal recognition models. We conclude with an examination of future directions.

2. Related Work

Recognizing players' goals and plans offers significant promise for increasing the effectiveness of digital game environments for education, training, and entertainment. Plan recognition, which seeks to infer agents' goals along with their plans for achieving them from sequences of observable actions, has been studied for tasks ranging from traffic monitoring (Pynadath and Wellman 2000) to operating system usage (Blaylock and Allen 2003) to story understanding (Charniak and Goldman 1993). Plan recognition is inherently uncertain, and solutions supporting reasoning under uncertainty such as Bayesian models (Charniak and Goldman 1993), probabilistic grammars (Pynadath and Wellman 2000), and Hidden Markov Models (Bui 2003) have demonstrated strong empirical performance in a range of settings. In the restricted form of plan recognition that focuses on inferring users' goals without concern for identifying their plans or sub-plans, goal recognition models have been automatically acquired using statistical corpus-based approaches without the need for hand-authored plan libraries (Blaylock and Allen 2003).

The classic goal recognition problem assumes that a single agent is pursuing a single goal using deterministic actions, and it assumes that a user's plan can be identified using a given plan library. A major focus of recent work on goal and plan recognition has been probabilistic approaches that relax several of these assumptions. For example, Ramirez and Geffner (2010) describe a plan recognition approach that does not require the provision of an explicit plan library. Hu and Yang (2008) describe a two-level goal recognition framework that uses conditional random fields and correlation graphs to support recognition of multiple concurrent and interleaving goals. Geib and Goldman (2009) have devised the PHATT algorithm, which is a Bayesian approach to plan recognition that focuses on plan execution. PHATT provides a unified framework that supports multiple concurrent goals, multiple instantiations of a single goal, and principled handling of unobserved actions. Geib and Goldman (2011) also proposed a lexical parsing-based approach to plan recognition that supports plans with loops. While probabilistic approaches have achieved considerable success, the propositional nature of probabilistic graphical models introduces limitations in formalizing plan recognition models.

Recent efforts have begun to focus on statistical relational learning frameworks for plan and activity recognition. Statistical relational learning techniques combine the representational strengths of logical formalisms and capabilities for reasoning under uncertainty, which are often associated with probabilistic graphical models. Of particular note, Markov Logic networks provide a single, unified formalism that supports structured representations and probabilistic reasoning. Sadilek and Kautz (2010) used Markov logic to investigate activity recognition in a multi-agent Capture the Flag game using GPS data. In their model, a Markov logic network combines hard and soft constraints derived from Capture the Flag rules to de-noise and label GPS data in terms of "capture" events. Experiments demonstrated that the MLN model significantly outperforms alternate probabilistic and nonprobabilistic approaches, correctly identifying 90% of capture events. While they are related, our work differs from Sadilek and Kautz's in several ways: (1) our models encode minimal domain-specific knowledge (i.e., our constraints are not first-order encodings of game rules), (2) we focus only on a single agent's behavior, and (3) we focus on modeling goals with cyclical relationships between goals and actions, rather than modeling players' activities.

Singla and Mooney (2011) devised a method for constructing MLN-based plan recognition models using abductive reasoning over planning domains. Their method is comprised of two parts: a Hidden Cause model, which augments Markov logic to support efficient abduction (inference from observations to explanations), and an abductive model construction procedure that reduces the number of atomic groundings considered while constructing ground Markov networks. Experiments found that Singla and Mooney's approach improved predictive accuracy over competing techniques. However, by framing the problem in terms of abductive inference, their approach requires that a formal description of the planning domain be available. In our work, goal recognition is conceptualized as a classification problem. Our approach does not directly perform abductive inference over the planning domain as in Singla and Mooney's approach, and therefore it does not required a planning domain to be explicitly provided.

There have been several studies of goal and plan recognition models in digital games. Kabanza and colleagues (2010) explored challenges with behavior recognition in real-time strategy (RTS) games, which involve opposing armies battling for control of regions using complex tactics and units. Their work extended Geib and Goldman's PHATT algorithm (2009) to perform intent recognition on the behaviors of RTS opponents. Preliminary experiments using game replays achieved 89% accuracy, although the authors assumed fully observable actions for the purposes of their initial evaluation. In other work, Laviers and Sukthankar (2011) combined plan recognition and plan repair models for real-time opponent modeling in a football videogame. Their approach used Upper Confidence Bounds applied to Trees (UCT) in order to learn plan repair policies in real-time. The plan repair policies enabled the offense to adjust its passing plan after recognizing the defense's play. In an empirical evaluation, the UCT approach outperformed baseline and heuristic-based plan repair methods in terms of yardage gain metrics and reductions in interceptions thrown. It should be noted that while both of these studies focused on plan recognition (and related) tasks in digital games, the tasks have major structural and application differences from our work involving exploratory actions in open-ended games.

More closely related to the study described in this chapter, Albrecht, Zukerman, and Nicholson (1998) conducted an early examination of Bayesian networks for plan recognition in adventure games. The study compared several alternate Bayesian network structures for predicting players' quests, actions, and locations in a text-based multi-user dungeon adventure game. The evaluation found that network structures incorporating both action and location data outperformed ablated versions with reduced connectivity and variables. However, no comparisons to non-DBN baselines were performed. More recently, Gold (2010) investigated Input-Output Hidden Markov Models (IOHMM) for recognizing high-level player goals in a simple 2D action-adventure game. The IOHMM model, which utilized hand-engineered parameters as opposed to machine-learned parameters, was compared to a hand-authored finite state machine (FSM), a common representational technique used in commercial games. A preliminary evaluation involving human subjects observed that the IOHMM model did outperform the FSM baseline. However, the generalizability of the findings was unclear due to the hand-authored nature of the model and baseline, as well as the relative simplicity of the application. The work also differed from our work in that it did not involve exploratory actions; player goals were directly presented to players in the upper left part of their screens. Studies of *The Restaurant Game* by Orkin (2007) have examined data-driven techniques for devising *plan networks* from thousands of user interactions with a simulated restaurant scenario. The plan networks encode common action sequences associated with customer-waiter interactions in a restaurant setting. An evaluation of the plan networks found they achieved high inter-rater reliability with human judges in assessing the 'typicality' of observed restaurant sequences in the game environment. While work on The Restaurant Game is related to classical plan recognition tasks, it has largely focused on different, and in some ways simpler, modeling tasks.

The work presented in this chapter investigates a Markov logic network goal recognition framework for an open-ended educational game, CRYSTAL ISLAND. CRYSTAL ISLAND shares several characteristics with adventure games, such as exploring a virtual environment, advancing an emerging narrative, and collecting and manipulating objects. While several goal recognition studies have been conducted using digital games, fundamental differences among the game genres and goal recognition tasks suggest that there is no established state-of-the-art performance level for this area. For this reason, we will evaluate the efficacy of our approach by comparing it against techniques employed in prior goal recognition work with an earlier version of our game environment.

3. Observation Corpus

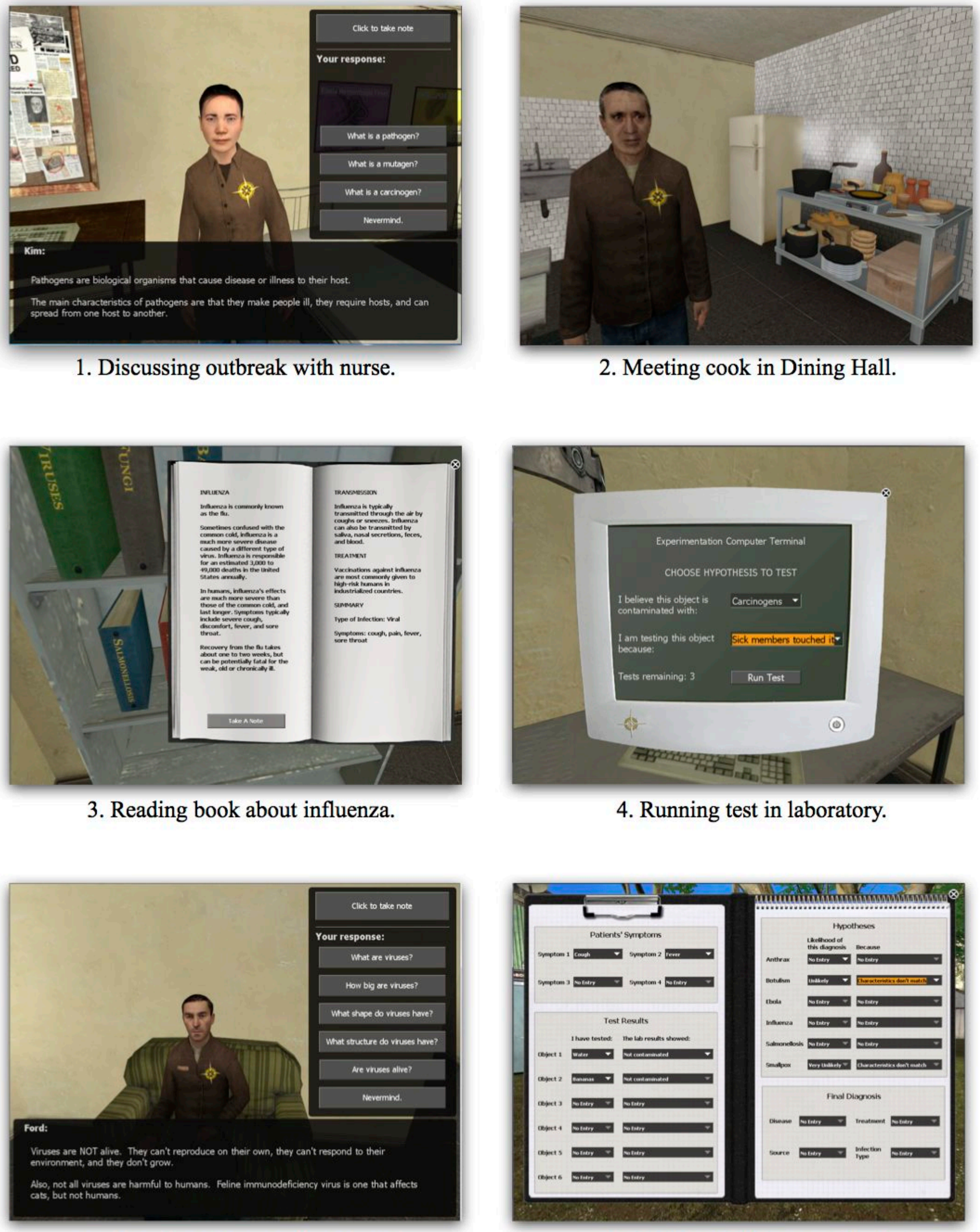
In order to investigate goal recognition in an open-ended game environment involving many possible goals and user actions, data collected from player interactions with the CRYSTAL ISLAND learning environment were used. CRYSTAL ISLAND (Figure 1) is a game-based learning environment for eighth-grade microbiology. It is built on Valve Software's Source^{TM} engine, the 3D game platform for Half-Life 2. The environment features a science mystery where players attempt to discover the identity and source of an infectious disease that is plaguing a research team stationed on an island. Players adopt the role of a visitor who recently arrived in order to see her sick father, but they are promptly drawn into a mission to save the entire research team from the outbreak. Players explore the research camp from a first-person viewpoint and manipulate virtual objects, converse with characters, and use lab equipment and other resources to solve the mystery. Players choose among multiple paths to investigate the mystery, and they are free to identify and pursue goals in a range of different orders.

CRYSTAL ISLAND has been the subject of extensive empirical investigation, and has been found to provide substantial learning and motivational benefits (Rowe et al. 2010). Players consistently demonstrate significant learning gains after using CRYSTAL ISLAND, and they report experiencing boredom less frequently than in alternative instructional software. CRYSTAL ISLAND is also challenging for players, with fewer than 50% of players solving the

Figure 1. C*RYSTAL* I*SLAND virtual environment.*

mystery in less than an hour. The current investigation of goal recognition models is part of an overarching research agenda that is focused on AI techniques for dynamically shaping players' interactions with game-based learning environments. Prior work has focused on a range of computational modeling tasks, including probabilistic representations for user knowledge modeling (Rowe and Lester 2010) and machine learning frameworks for driving characters' affective behaviors (Robison, McQuiggan, and Lester 2009). The present work is most closely related to two prior investigations of user models with CRYSTAL ISLAND. The first, and closest, relation is Mott and Lester's investigation of Bayesian networks and n-gram models for goal recognition (2006). While we adopt and extend several ideas from that research, the present work uses a significantly modified version of CRYSTAL ISLAND, as well as a version that does not explicitly advise players about which goals to next achieve. The current version of CRYSTAL ISLAND is substantially more open-ended, as players are encouraged to identify and follow their own paths to solve the mystery. The present work also compliments a prior investigation of user knowledge modeling with dynamic Bayesian networks (Rowe and Lester, 2010). That work employed hand-engineered dynamic Bayesian networks to model player beliefs related to the narrative backstory, science curriculum, mystery solution, and gameplay mechanics. The current work models users' goals with a Markov logic network, which leverages parameters (i.e., weights) induced from user interaction data. The two categories of models are distinct but complimentary, and in concert they could be used to drive decisions about how to dynamically tailor game events to individual players.

Player interactions with CRYSTAL ISLAND are comprised of a diverse set of actions occurring throughout the seven major locations of the island's research camp: an *infirmary*, a *dining hall*, a *laboratory*, a *living quarters*, the *lead scientist's quarters*, a *waterfall*, and a large *outdoors* region. Players can perform actions that include the following: *moving around the camp*, *picking up* and *dropping objects*, *using the laboratory's testing equipment*, *conversing with virtual characters*, *reading microbiology-themed books* and *posters*, *completing a diagnosis worksheet*, *labeling microscope slides*, and *taking notes*. Players advance through CRYSTAL ISLAND's non-linear narrative by completing a partially ordered sequence of goals that comprise the scenario's plot. Seven narrative goals are considered in this work: *speaking with the camp nurse* about the spreading illness, *speaking with the camp's virus expert*, *speaking with the camp's bacteria expert*, *speaking with a sick patient*, *speaking with the camp's cook* about recently eaten food, *running laboratory tests on contaminated food*, and *submitting a complete diagnosis* to the camp nurse.



Kim:

Pathogens are biological organisms that cause disease or illness to their host.

The main characteristics of pathogens are that they make people ill, they require hosts, and can spread from one host to another.

1. Discussing outbreak with nurse.

2. Meeting cook in Dining Hall.

INFLUENZA

Influenza is commonly known as the flu.

Sometimes confused with the common cold, influenza is a much more severe disease caused by a different type of virus. Influenza is responsible for an estimated 3,000 to 49,000 deaths in the United States annually.

In humans, influenza's effects are much more severe than those of the common cold, and last longer. Symptoms typically include severe cough, discomfort, fever, and sore throat.

Recovery from the flu takes about one to two weeks, but can be potentially fatal for the weak, old or chronically ill.

TRANSMISSION

Influenza is typically transmitted through the air by coughs or sneezes. Influenza can also be transmitted by saliva, nasal secretions, feces, and blood.

TREATMENT

Vaccinations against influenza are most commonly given to high-risk humans in industrialized countries.

SUMMARY

Type of Infection: Viral

Symptoms: cough, pain, fever, sore throat

3. Reading book about influenza.

Experimentation Computer Terminal

CHOOSE HYPOTHESIS TO TEST

I believe this object is contaminated with: Carcinogens

I am testing this object because: Sick members touched it

Tests remaining: 3

4. Running test in laboratory.

Ford:

Viruses are NOT alive. They can't reproduce on their own, they can't respond to their environment, and they don't grow.

Also, not all viruses are harmful to humans. Feline immunodeficiency virus is one that affects cats, but not humans.

5. Discussing viruses with scientist.

Patients' Symptoms

Symptom 1	Cough	Symptom 2	Fever
Symptom 3	No Entry	Symptom 4	No Entry

Test Results

	I have tested:	The lab results showed:
Object 1	Water	Not contaminated
Object 2	Bananas	Not contaminated
Object 3	No Entry	No Entry
Object 4	No Entry	No Entry
Object 5	No Entry	No Entry
Object 6	No Entry	No Entry

Hypotheses

	Likelihood of this diagnosis	Because
Anthrax	No Entry	No Entry
Botulism	Unlikely	Characteristics don't match
Ebola	No Entry	No Entry
Influenza	No Entry	No Entry
Salmonellosis	No Entry	No Entry
Smallpox	Very Unlikely	Characteristics don't match

Final Diagnosis

Disease	No Entry	Treatment	No Entry
Source	No Entry	Infection Type	No Entry

6. Recording findings in worksheet.

Figure 2. Sample events in the CRYSTAL ISLAND narrative.

The following scenario illustrates a typical interaction with CRYSTAL ISLAND (see Figure 2 for a related sequence of screenshots). The scenario begins with the player's arrival at the research camp (Figure 3). The player approaches the first building, an infirmary, where several sick patients and a camp nurse are located. The player approaches the nurse and initiates a conversation with her. The nurse explains that an unidentified illness is spreading through the camp and asks for the player's help in determining a diagnosis. The conversation with the nurse takes place through a combination of multimodal character dialogue—spoken language, gesture, facial expression, and text—and player dialogue menu selections. All character dialogue is provided by voice actors and follows a deterministic branching structure.

After speaking with the nurse, the player has several options for investigating the illness. Inside the infirmary,

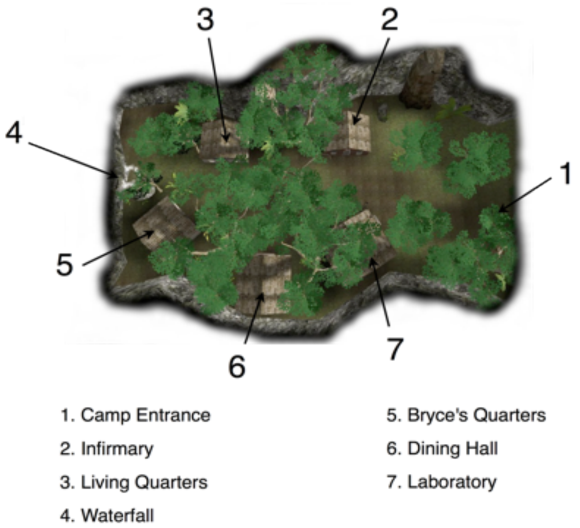


Figure 3. Map of the CRYSTAL ISLAND research camp.

the player can talk to sick patients lying on medical cots. Clues about the team members' symptoms and recent eating habits can be discussed and recorded using in-game note-taking features. Alternatively, the player can move to the camp's dining hall to speak with the camp cook. The cook describes the types of food that the team has recently eaten and provides clues about which items warrant closer investigation. In addition to learning about the sick team members, the player has several options for gathering information about disease-causing agents. For example, the player can walk to the camp's living quarters where she will encounter a pair of virtual scientists who answer questions about viruses and bacteria. The player can also learn more about pathogens by viewing posters hanging inside of the camp's buildings or reading books located in a small library. In this way, the player can gather information about relevant microbiology concepts using resources that are presented in multiple formats.

Beyond gathering information from virtual scientists and other instructional resources, the player can conduct tests on food objects using the laboratory's testing equipment. The player encounters food items in the dining hall and laboratory, and she can test the items for pathogenic contaminants at any point during the learning interaction. A limited number of tests are allocated to the player at the start of the scenario, but additional tests can be earned by answering microbiology-themed multiple-choice questions posed by the camp nurse.

After running several tests, the player discovers that the sick team members have been consuming contaminated milk. Upon arriving at this finding, the player is instructed to see the lab technician, Elise, for a closer look. The screen momentarily fades to black to indicate elapsing time, and Elise returns with an image of the contaminated specimen, which she explains was taken using a microscope. At this point, the player is presented with a labeling exercise where she must identity the contamination as bacterial in nature. After successfully completing this activity, the player uses in-game resources to investigate bacterial diseases that are associated with symptoms matching those reported by the sick team members. Once she has narrowed down a diagnosis and recommended treatment, she records them in a diagnosis worksheet and returns to the infirmary to inform the camp nurse. If the player's diagnosis worksheet contains errors, the nurse identifies the errors and recommends that the player keep working. If the player correctly diagnoses the illness and specifies an appropriate treatment, the mystery is solved.

The data used for investigating MLN goal recognition models were collected from a study involving the eighth grade population of a public middle school. There were 153 participants in the study. Data for sixteen of the participants was removed from the analysis due to incomplete data or prior experience with CRYSTAL ISLAND. Participants whose data was included had no prior experience with the software. All player actions were logged by the CRYSTAL ISLAND software and stored for later analysis. The log files were generated by a modified version of the Source^{TM} engine's built-in logging features, and they consist of chronological, flat sequences of time-stamped actions. In the corpus, each student is associated with a single log file, except in cases of hardware/software crashes, which occasionally result in multiple logs. Examples of actions recorded in the log files include movements between locations, pickup object events, drop object events, conversation initiation events, dialog turn events, read book events, and goal achievement events. Additionally, the raw logs include records of players' positions and orientations, which are recorded several times per second, although the data is filtered out in the present analysis.

Arguments for each action (e.g., the object that was dropped, the book that was read) are included in each entry of the logs.

4. Markov Logic Networks

The goal recognition model described in this chapter utilizes the statistical relational learning framework provided by *Markov logic networks* (MLNs) (Richardson and Domingos, 2006). Statistical relational learning frameworks support modeling complex phenomena for domains in which the assumption of *independent and identically distributed (i.i.d)* data does not hold. This section provides a brief overview of MLN.

An MLN is a knowledge base that consists of first-order logic formulae and a finite set of constants representing domain objects. In contrast to traditional first-order logic, in which logic formulae have binary values (i.e., *true* or *false*) and represent hard constraints, each formula in an MLN is associated with a weight that has a real number value and represents a soft constraint that is allowed to be violated. When the arguments of every predicate in the first-order knowledge base are replaced with all possible domain constants and a truth value is assigned to each ground predicate, it represents a possible world that can be constructed from the given knowledge base. In traditional first-order logic, possible worlds have binary values: the value of a possible world is *true* when all the ground formulae in the knowledge base are true; otherwise, it is *false*. In contrast, an MLN defines a probability distribution over possible worlds in a continuous range. Weights of MLN formulae reflect the strength of the constraint that their associated logic formula imposes on the possible worlds. Table 1 shows an example of a simple MLN, which consists of two first-order logic formulae and associated weights.

First-Order Logic	Clausal Form	Weight
$\forall x\ Cold(x) \Rightarrow Coughs(x)$ If a person has a cold then they cough.	$\neg Cold(x) \lor Coughs(x)$	1.5
$\forall x \forall y\ Family(x,y) \Rightarrow (Cold(x) \Leftrightarrow Cold(y))$ If two people are family, they both have colds or neither does.	$\neg Family(x,y) \lor Cold(x) \lor \neg Cold(y)$ $\neg Family(x,y) \lor \neg Cold(x) \lor Cold(y)$	1.1 1.1

Table 1. Example MLN

Formally, an MLN is defined as a set of tuples (f_i, w_i), in which f_i is a first-order logic formula in the knowledge base and w_i is a real number. Grounded with domain constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, an MLN defines a *Markov network* (MN), also know a *Markov random field*. An MN is an undirected probabilistic graphical model that defines a joint distribution of a set of random variables $X = (X_1, X_2, \ldots, X_n) \in \chi$ (Pietra et al., 1997). The MN defined by a ground MLN has a graph node for each ground predicate in the MLN. The value of the node is 1 if the corresponding ground predicate is *true* and 0 otherwise. Two nodes in the graph are connected by an edge if and only if their corresponding ground predicates appear together in at least one ground MLN formula. Thus, the MN has a unique feature (i.e., a clique in the graph) for each ground MLN formula. Figure 4 depicts a graphical structure of a ground MLN, which is constructed by applying domain constants A and B to the MLN in Table 1. The value of the feature is 1 if the corresponding ground MLN formula is *true* and 0 otherwise. Features in MNs are assigned the same weights as their corresponding MLN formulae. The probability distribution over possible worlds α for a ground MLN is defined by the following equation:

$$P(X = \alpha) = \frac{1}{Z} \exp\left(\sum_{i=1}^{F} w_i n_i(\alpha)\right), \qquad (1)$$

in which F represents the number of formulae in the MLN and $n_i(\alpha)$ is the number of groundings of a formula f_i that has *true* value in the given world α. Z is a normalization constant, which is computed as:

$$Z = \sum_{x} \exp\left(\sum_{i=1}^{F} w_i n_i(x)\right). \qquad (2)$$

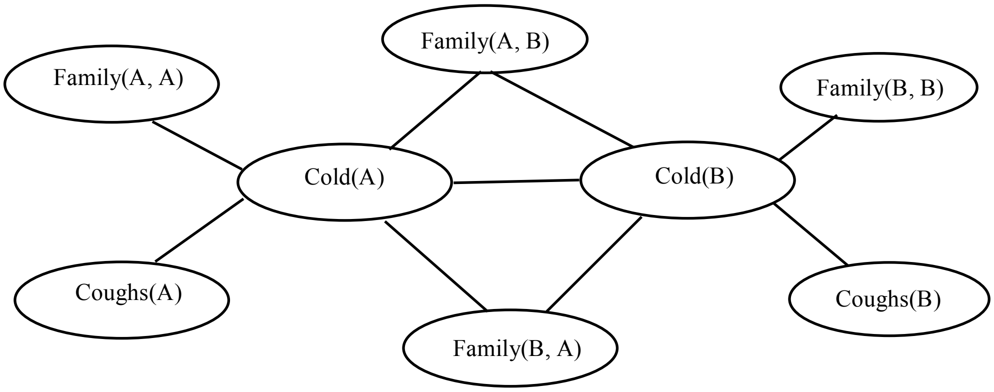


Figure 4. Graphical structure of MLN in Table 1

Cutting Plane Inference (CPI) provides an accurate and efficient Maximum A Posteriori (MAP) method for MLNs (Riedel, 2008). CPI is a meta-algorithm that instantiates a small fraction of a given complex MLN incrementally and solves it using a conventional MAP inference method such as MaxWalkSAT or integer linear programming. At each incremental step, the CPI refines a current solution by instantiating only those portions of the complete MN for which the current solution can be further optimized. CPI has been shown to improve the efficiency of MAP inference compared to conventional methods alone.

By combining first-order logic syntax and probabilistic graphical models, MLNs bring together the characteristics of both probabilistic and logical approaches to modeling complex phenomena. From a probability perspective, MLNs can be viewed as templates for constructing large Markov networks, which allow an incremental, modular approach to incorporate knowledge bases into the network. From a logic perspective, Markov logic provides a mechanism to handle uncertainty and contradictory data in a knowledge base. The work presented in this chapter leverages MLNs as a template language to construct a Markov network for goal recognition in open-ended digital games.

5. Goal Recognition with Markov Logic Networks

Drawing upon previous goal recognition research, (Blaylock and Allen 2003; Mott et al. 2006), the work presented in this chapter defines goal recognition as the task of predicting the most likely goal for a given sequence of observed low-level player behavior in the game environment. It is assumed that a given sequence of player behavior maps to a single goal, and no interleaving occurs between different goals. We attribute all actions leading up to a goal (and taking place after the previous goal was achieved) as part of the action sequence associated with that goal. In part, this assumption is due to the nature of our data: during the data collection, we could not directly observe students' intentions as they played CRYSTAL ISLAND. Therefore, we identified students' narrative goals during post hoc analysis. During the data collection, students did not self-report their goals, nor did trained human experts provide judgments about students' intentions as they played CRYSTAL ISLAND. While these techniques—student think-alouds and expert observational protocols—hold promise for yielding data that enable relaxation of single-goal assumptions, these efforts are reserved for future work.

Under the described conditions, goal recognition is cast as a classification problem, in which a learned classifier predicts the most likely goal for a currently observed player behavior. With the given task formulation, the current goal recognition model leverages task structure in CRYSTAL ISLAND both at the local and global levels: the local task structure is encoded as pairwise ordering patterns of goals, and the global task structure is incorporated by a special attribute that encodes the player's progress in solving the problem within the narrative scenario. This section discusses the representation of player behavior used in our goal recognition model (Section 5.1), the motivation to use MLNs for goal recognition (Section 5.2), and the MLN goal recognition model (Section 5.3).

5.1 Representation of Player Behavior

Similar to previous work by Mott, Lee, and Lester (2006), the current work encodes low-level player behavior in the game environment using three attributes: *action type*, *location*, and *narrative state*.

- **Action Type:** Type of current action taken by the player, such as *moving to a certain location*, *opening a door*, and *testing an object using the laboratory's testing equipment*. Our data includes 19 distinct types of player actions. The current work only considers the type (e.g., *OPEN*) of the action but does not include the associated arguments (e.g., *laboratory-door*) because adding action arguments did not offer clear benefits for the learned model, which was likely caused by a data sparsity problem. However, action arguments in principle provide richer semantics for goal recognition models. In future work, we plan to investigate the utility of action arguments with a larger data set.

- **Location:** Place in the virtual environment where a current player action is taken. This includes 39 fine-grained and non-overlapping sub-locations that decompose the seven major camp locations in CRYSTAL ISLAND.
- **Narrative state:** Representation of the player's progress in solving the narrative scenario. Reflecting the global task structure of CRYSTAL ISLAND, narrative state is encoded as a vector of four binary variables, each of which represents a milestone event within the narrative. The four milestone events considered are: *Discuss the illness with the nurse*, *Test the contaminated object, Submit a diagnosis to the nurse,* and *Submit a correct diagnosis to the nurse*. The first two of these are also subgoals that the player needs to achieve. If a milestone event has been accomplished, the associated variable is assigned a value of 1. Otherwise the value of the variable is 0.

5.2 Motivation

The data described in the previous section poses significant challenges for goal recognition. First, individual goals are not independent of one another. Goals in our data represent milestone activities players take in the course of solving the science mystery. Some of these activities naturally occur in sequential patterns driven by the game's narrative structure. The layout of the island can also impose co-occurrence patterns among goals. Thus, the previous goal can impact the probabilities associated with alternate values for the current goal. To model these associations among the milestone activities, goals should be inferred in relation to one another, rather than in isolation. Second, the causality between player behavior and goals is ambiguous. In CRYSTAL ISLAND, players are not given an explicit list of goals to achieve. Instead, players discover goals while interacting with the virtual environment. Thus, causality between player behavior and goals is bidirectional: a goal could influence a player's current behavior if she has a particular goal in mind, and it is also possible that the player's current behavior will reveal which goal she next pursues. For instance, a player can enter a new location without a particular goal in mind, and afterward she can engage a character in conversation, which reveals a new goal.

To address the first challenge, the current work utilizes a statistical relational learning framework. Statistical relational learning frameworks offer principled methods for collective classification, in which multiple objects are jointly modeled (Getoor and Taskar 2007). In our case, the goal recognition model jointly classifies successive goals. To address the second challenge, the current work particularly considers MLNs (Richardson and Domingos 2006), because of the underlying semantics of undirected graphical models (Section 4). In contrast to directed graphical models, which encode explicit causality among the entities, undirected graphical models represent mutual influence among entities, which is well suited for representing ambiguous causality between player behavior and goals in our data. In addition, the first-order logic syntax of MLNs offers intuitive interface for developers to work with data in order to construct complex Markov networks.

5.3 Markov Logic Network for Goal Recognition

We first defined a set of predicates as the basic building blocks to form MLN formulae for the proposed goal recognition model. There are two types of predicates, *observed* and *hidden*. Observed predicates are those that are fully observable by the game environment while a player is performing actions. In contrast, hidden predicates are those that are not directly observable by the game environment. Instead, the groundings of the hidden predicates are predicted using MAP inference based on a learned model. In other words, hidden predicates represent the target phenomena that are being modeled. In our case, there is one hidden predicate, *goal(t, g),* which represents the player's goal at time *t*. Three observed predicates were defined, each representing an attribute of player behavior: *action(t, a)*, *loc(t, l)*, and *state(t, s)*. Table 2 lists these observed and hidden predicates.

Predicate		Description
Observed	*action(t, a)*	Player takes action *a* at time *t*.
	loc(t, l)	Player is at location *l* at time *t*.
	state(t, s)	The narrative state at time *t* is *s*.
Hidden	*goal(t, g)*	Player pursues goal *g* at time *t*.

Table 2. Observed and Hidden Predicates

By combining the observed and the hidden predicates with logical operations, a total of 13 MLN formulae were constructed. As shown in Figure 5, our goal recognition MLN consists of one hard formula and 12 soft formulae. A hard formula represents a constraint that needs to be satisfied at all times. For instance, Formula F1 requires that, for each action *a* at each time step *t*, there exists a single goal *g*. In MLNs, hard formulae are assigned an arbitrarily large weight. The formulae F2-F13 are soft formulae, which represent constraints that are allowed to be violated. Formula F2 reflects prior distribution of goals in the corpus. Formula F3-F11 predict a player's goal *g* at a given time *t* based on the observed player behavior. Instead of aggregating all three attributes of player behavior in a single formula, we modeled the impacts of each individual attribute separately by defining a MLN formula for each attribute. Combinations of the attributes were also considered. In contrast to Formula 3-11 that predict individual goal separately from others, formulae F12 and F13 jointly predict sequentially adjacent pairs of goals by exploiting their ordering patterns. The weights for the soft constraints were learned by using *theBeast*, an off-the-shelf tool for MLNs that employs the CPI method for MAP inference (Riedel 2008). All of these formulae are included in the current goal recognition model. Figure 6 illustrates a partial graphical representation of the described MLN.

Hard Formula

$$\forall t,a : action(t,a) \Rightarrow \left|\exists g : goal(t,g)\right| = 1 \qquad \text{(F1)}$$

Soft Formulae

$$\forall t,g : goal(t,g) \qquad \text{(F2)}$$

$$\forall t,a,g : action(t,a) \Rightarrow goal(t,g) \qquad \text{(F3)}$$

$$\forall t,l,g : loc(t,l) \Rightarrow goal(t,g) \qquad \text{(F4)}$$

$$\forall t,s,g : state(t,s) \Rightarrow goal(t,g) \qquad \text{(F5)}$$

$$\forall t,a,s,g : action(t,a) \wedge state(t,s) \Rightarrow goal(t,g) \qquad \text{(F6)}$$

$$\forall t,a,g : action(t-1,a) \Rightarrow goal(t,g) \qquad \text{(F7)}$$

$$\forall t,l,g : loc(t-1,l) \Rightarrow goal(t,g) \qquad \text{(F8)}$$

$$\forall t,s,g : state(t-1,s) \Rightarrow goal(t,g) \qquad \text{(F9)}$$

$$\forall t,a,s,g : action(t-1,a) \wedge state(t-1,s) \Rightarrow goal(t,g) \qquad \text{(F10)}$$

$$\forall t,a_1,a_2,g : action(t-1,a_1) \wedge action(t,a_2) \Rightarrow goal(t,g) \qquad \text{(F11)}$$

$$\forall t,g_1,g_2 : goal(t-1,g_1) \Rightarrow goal(t,g_2) \qquad \text{(F12)}$$

$$\forall t,a_1,a_2,g_1,g_2 : action(t-1,a_1) \wedge goal(t-1,g_1) \wedge action(t,a_2) \Rightarrow goal(t,g_2) \qquad \text{(F13)}$$

Figure 5. Formulae for MLN goal recognition model

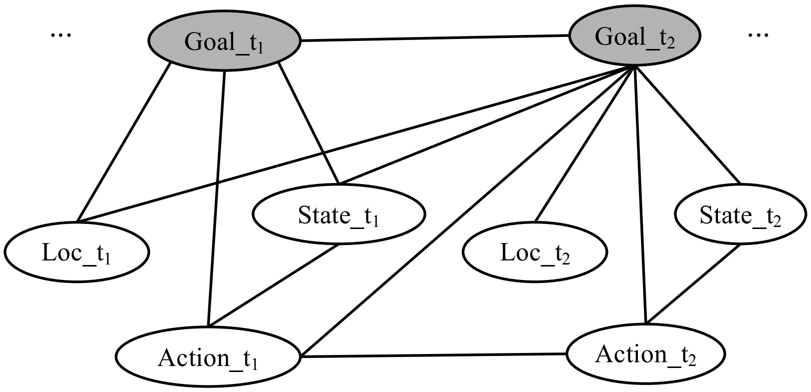


Figure 6. Graphical Representation of Goal Recognition MLN

6. Evaluation

To train and test the goal recognition MLN, the data from the observation corpus was automatically preprocessed in several steps. First, all player actions that achieve goals were identified. Because the goals considered in our work are a set of milestone actions players achieved while solving the narrative scenario, they are logged in the same manner as other actions. Therefore, identification of goal-achieving actions can be performed automatically by scanning the gameplay data, without need for manually defining domain-specific knowledge, such as which actions correspond to which goals. Second, all actions in the observation sequence that precede the current goal but follow the previous goal were labeled with the current goal. Third, actions that achieve goals were removed from the data. Removing goal-achieving actions was necessary to ensure that model training was fair, because it would be trivial to predict goals from the goal-achieving actions. Finally, all actions that were taken after achievement of the last goal were also removed, since those actions did not lead to achieving a goal. Table 3 shows summary statistics of the resulting corpus, which includes 77,182 player actions and 893 achieved goals, with an average of 86.4 player actions per goal. Table 4 shows the set of goals considered in this work and their frequencies in the processed corpus data. The most frequent goal was *Running laboratory test on contaminated food*, which comprised 26.6% of the total goals in the corpus.

Total number of observed player actions	77182
Total number of goals achieved	893
Average number of player actions per goal	86.4

Table 3. Statistics for observed actions and goals.

Running laboratory test on contaminated food	26.6%
Submitting a diagnosis	17.1%
Speaking with the camp's cook	15.2%
Speaking with the camp's bacteria expert	12.5%
Speaking with the camp's virus expert	11.2%
Speaking with a sick patient	11.0%
Speaking with the camp nurse	6.4%

Table 4. Distribution of goals.

For evaluation, the proposed MLN model was compared to a trivial baseline system and two non-trivial baselines. The trivial baseline was the majority baseline, which always predicted the goal that appears most frequently in the training data. The non-trivial baselines were two *n*-gram models, adopted from previous goal recognition work (Blaylock and Allen 2003; Mott et al. 2006). Although simplistic, *n*-gram models have been shown to be effective for goal recognition in interactive narrative game environments (Mott et al. 2006) and spoken dialogue systems

(Blaylock and Allen 2003). The following sections provide a detailed description of the *n*-gram comparison systems (Section 6.1) and report evaluation results (Section 6.2).

6.1 *N*-gram Models

Previous work by Mott et al. (2006) defined that, given an observation sequence $O_1, O_2, \ldots, O_n$, the objective of goal recognition for interactive narrative environment is to identify the most likely goal G^*such that:

$$G^* = \arg\max P(G \mid O_1, O_2, \ldots, O_n) = \arg\max P(G \mid O_{1:n}), \tag{3}$$

where O_i is an observation of player behavior at a time step *i*. Applying Bayes' rule followed by the chain rule, the equation (3) becomes:

$$G^* = \arg\max P(O_n \mid O_{1:n-1}, G) P(O_{n-1} | O_{1:n-2}, G) \ldots P(O_1 | G) P(G). \tag{4}$$

Since estimating these conditional probabilities is impractical, following an earlier approach (Blaylock and Allen 2003), Mott et al. (2006) defined two *n*-gram models based on a Markov assumption that an observation O_i depends only on the goal G and a limited window of preceding observations. The unigram model states that given the goal G, O_i is conditionally independent of all other observations (Equation 5). Thus, The unigram model predicts a goal based on the current observation of player behavior only. Extending the window size by one time step, the bigram model assumes that, given the goal G and a preceding observation O_{i-1}, O_i is conditionally independent of all other observations (Equation 6). Therefore, the bigram model considers the previous observation of player behavior as well. An observation O_i for the *n*-gram models is an aggregate variable that combines all the attributes of player behavior. In spite of the simplistic approach, Mott et al. (2006) found *n*-gram models achieved higher prediction accuracies than a more sophisticated Bayesian network model. Similar to the MLN model described in Section 5, their Bayesian network model used factored representation of player behavior, modeling the influence of each attribute of player behavior separately.

$$G^* = \arg\max P(G) \textstyle\prod_{i=1}^{n} P(O_i | G) \tag{5}$$

$$G^* = \arg\max P(G) \textstyle\prod_{i=1}^{n} P(O_i | O_{i-1}, G) \tag{6}$$

Adopting these ideas within the MLN framework, in the work presented in this chapter we constructed a unigram and a bigram goal recognition models as MLNs, which employed an aggregate representation of player behavior. The unigram and the bigram models were defined by a single soft formula F14 and F15, respectively. Formula F14 predicts a goal at time *t* from the combination of all three attributes of player behavior observed at the same time step. Formula F15 combines the attributes of player behavior for time *t* as well as time *t-1*, in order to predict a goal at time *t*. In order to ensure that each player action at each time step is assigned a single goal, the hard formula F1 (Figure 5) was also added to both the unigram and the bigram models. Without any formula that encodes relations among different goals, *n*-gram models predict each individual goal separately and do not take into consideration the sequential association patterns among them.

$$\forall t, a, l, s, g: action(t, a) \wedge loc(t, l) \wedge state(t, s) \Rightarrow goal(t, g) \tag{F14}$$

$$\begin{aligned}\forall t, a_1, a_2, l_1, l_2, s_1, s_2, g: & \, action(t, a_1) \wedge loc(t, l_1) \wedge state(t, s_1) \\ & \wedge action(t-1, a_2) \wedge loc(t-1, l_2) \wedge state(t-1, s_2) \Rightarrow goal(t, g)\end{aligned} \tag{F15}$$

6.2 Results

The two *n*-gram models, which employed an aggregate representation of player behavior, and the proposed MLN model, which utilized a factored representation, were trained using one-best MIRA (Crammer and Singer 2003) as the update rule. The entire data set was partitioned into ten non-overlapping subsets, ensuring data from the same player did not appear in both the training and the testing data. Each subset of the data was used for testing exactly once. The three models were evaluated with ten-fold cross validation on the entire data set. The models' performance was measured using *F1*, which is the harmonic mean of *precision* and *recall*. In the case of multi-class classification such as ours, there are multiple approaches to computing *F1*. Common approaches include computing either the micro-average or macro-average of *F1*. Micro-average computes *F1* by treating all objects in the testing data equally regardless of the class. Macro-average takes the average of *F1* scores that are separately computed for each class, by treating multi-class classification as an aggregate of binary classification. A drawback of these approaches is entries in the confusion matrix are counted multiple times. In order to avoid this, an alternate approach uses the slightly different formulations of *precision* (equation 7) and *recall* (equation 8) for multi-class classification, in which *F1* score is computed as a harmonic mean of precision and recall as for the binary classification (e.g., Marcu 2000). Our work follows this approach. It should be noted that in the current work the values of *precision*, *recall*, and *F1* are the same, because each observed player action is associated with a single goal in our data and the goal recognition model predicts a single most likely goal for each player action (i.e., *total number of predictions made by classifier* is equal to *total number of objects*). These values again coincide with *accuracy*, another metric widely used for multi-class classification.

$$Precision = \frac{Total\ number\ of\ correctly\ classified\ objects}{Total\ number\ of\ predictions\ made\ by\ classifier} \quad (7)$$

$$Recall = \frac{Total\ number\ of\ correctly\ classified\ objects}{Total\ number\ of\ objects} \quad (8)$$

Table 5 shows the average performance of each compared model over ten-fold cross validation. The factored MLN model scored 0.484 in *F1*, achieving an 82% improvement over the baseline. The unigram model performed better than the bigram model. A one-way repeated-measures ANOVA confirmed that the differences among the three compared models were statistically significant ($F(2,18) = 71.87$, $p < 0.0001$). A post hoc Tukey test revealed the differences between all pairs of the three models were statistically significant ($p < .01$).

	Baseline	Unigram	Bigram	Factored MLN
F1	0.266	0.396	0.330	**0.484**
Improvement over Baseline	N/A	49%	24%	**82%**

Table 5. F1 scores for MLN and baseline goal recognition models.

7. Discussion

While all three models performed better than the baseline, the factored MLN model achieved the best performance, suggesting that the proposed MLN goal recognition approach is effective in predicting player goals from observed player behavior in a complex game environment. The *F1* score of 0.484 achieved by the factored MLN model may appear somewhat low. However, this result is encouraging given the challenges posed by the data. The chance probability of accurately recognizing the seven goals in our data is 0.143. The superiority of the factored MLN model compared to the *n*-gram models can be partially explained by the MLN's relational learning framework, which facilitates explicit modeling of associations between goals: the factored MLN model jointly predicts successive goals, while *n*-gram models predicted each individual goal separately. Furthermore, the factored representation of player behavior enables MLNs to model richer relations between player behavior and goals than the aggregate representation used in *n*-gram models. The finding that the unigram model achieved higher performance than the bigram model is consistent with the result reported by Mott et al. (2006). Among the possible reasons for the unigram model's superiority over the bigram model is data sparsity. The bigram model considers two consecutive previous goals, which would lead to greater sparsity than in the unigram model.

The current work is a first step toward a general approach for modeling exploratory goal recognition in digital games using MLNs. The current goal recognition model makes relatively basic use of MLNs, modeling only sequential relations among successive goals, but MLNs provide a principled method to incorporate richer structural relations present among the goals, such as the relations imposed by the narrative structure of the virtual world. Also, the current work is characterized as a knowledge-lean approach to goal recognition, which uses only a minimal amount of domain knowledge in the model's specification. The only domain knowledge required for the proposed goal recognition model was to identify the milestone events within the CRYSTAL ISLAND game environment. The knowledge-lean approach offers a cost-efficient solution for devising goal recognition models because identifying milestone events is relatively easy compared to manually annotating the data with a hand-crafted goal schema or hand-authoring the mappings between player behavior and goals. We also suspect that knowledge-lean approaches offer a more robust solution than competing approaches to goal recognition in open-ended games, due to challenges associated with unexpected patterns in players' exploratory behavior.

Inducing accurate goal recognition models has several prospective benefits for game-based learning environments. Goal recognizers can be used to inform player-adaptive decisions by *narrative-centered tutorial planners,* which comprise a particular class of software agents that simultaneously reason about interactive narrative and pedagogical issues to personalize student's game-based learning experiences. Data-driven approaches to narrative-centered tutorial planning are the subject of active research by the CRYSTAL ISLAND research team. They offer a method for dynamically tailoring events during players' game-based learning experiences in order to individualize pedagogical scaffolding and promote player engagement. Consider the following scenario: a player begins running laboratory tests prematurely, collecting data before he has gathered enough background information about CRYSTAL ISLAND's outbreak scenario to adequately interpret the results. Goal recognition models enable CRYSTAL ISLAND to predict that the player is attempting to find a contaminated object. However, this prediction prompts a subsequent inference that the player's goal is sub-optimal for the current stage of the narrative. This realization triggers an event where a virtual character suggests that the player should instead go speak with sick patients to better understand how the disease spread through the research camp. In this manner, a narrative-centered tutorial planner can dynamically tailor events to improve students' learning outcomes and narrative experience, driven by information from goal recognition models.

Similarly, goal recognition models can be employed to detect situations where players attempt to solve the mystery by "gaming the system" (Baker, Corbett, Koedinger, and Wagner, 2004). In some cases, players may attempt to solve the mystery by guessing as they fill out the diagnosis worksheet. In this exploitative strategy, players guess responses on the worksheet, submit it to the camp nurse, receive feedback about incorrect responses, revise the worksheet, and repeat the process in hope of stumbling upon the correct solution. A goal recognition model could predict that the player is attempting to solve the mystery. However, if the player has not yet completed other important narrative milestones, it might be appropriate for the camp nurse to prompt the player to consider an alternate course of action. The nurse might prompt the player to speak with other scientists on the island, or collect data in the virtual laboratory, in order to obtain knowledge that will enable the player to complete the diagnosis worksheet in a more deliberative fashion.

In addition to using goal recognition models to dynamically shape events in CRYSTAL ISLAND's narrative, goal recognizers can be used during data mining to inform the analysis and design of future iterations of the CRYSTAL ISLAND software. By automatically recognizing players' goals, and identifying which actions are likely to be associated with those goals, the research team can gain insights about common types of problem-solving paths and challenges encountered by players. Furthermore, recognizing players' goals can enrich in-game assessments of player learning and problem solving, which is a critical challenge for the serious games community.

The present work has several limitations that we seek to address through continued investigation. First, the available log data did not include pre-existing class labels of players' intentions. To investigate goal recognition, it was necessary to add class labels during subsequent data analysis, which necessitated single-goal assumptions that prevented consideration of concurrent or interleaving goals. In future data collections with CRYSTAL ISLAND, it may be possible to obtain class labels, or the information necessary to extract class labels, during run-time. A common method for obtaining such data in educational research is to ask students to think aloud as they solve the problem, often with support from a researcher providing regular prompts. While this approach provides a direct window into students' intentions, in the past we have seen considerable variability in middle grade students' willingness to verbalize their thoughts as they play CRYSTAL ISLAND. This may be due to cognitive difficulties associated with simultaneously playing an unfamiliar game and verbalizing one's own intentions, or it may be a symptom of adolescent resistance to verbal communication. In the case of the former, it may be possible to circumvent this challenge by having students watch video recordings of their own gameplay sessions after solving the mystery, providing verbal commentary about their own play experiences in a post hoc manner. An alternative approach is training human "experts" to provide judgments about player intentions by observing students as they play the game. This approach depends on humans accurately recognizing students' intentions in CRYSTAL ISLAND; in this case goal data is not directly obtained from players. On the one hand, this method bypasses the need for students to verbalize their own intentions, which is challenging in practice. On the other hand, it is resource-intensive to gather human judgments for a large number of student gameplay sessions, a problem only exacerbated by the need for multiple judgments to enable inter-rater reliability calculations.

The above methods, think-alouds and expert observational protocols, also introduce opportunities for addressing another limitation of the present work: detecting non-traditional goals that are not represented in the set of seven narrative-sub goals. For example, students may adopt ill-defined goals, such as "explore the virtual environment", or non-narrative goals, such as "climb on top of the dining hall," to drive their actions. Accurately predicting these goals would substantially extend the capabilities of narrative-centered tutorial planners seeking to interpret player actions and tailor game events to individual students. However, it is necessary to obtain class labels to model these goals using the MLN-based classification approach presented in this work. Furthermore, plan recognition methods that rely on inference over planning domains are unlikely to be effective for these types of intentions, as they most likely exceed the specifications typically included in most planning domain knowledge bases.

8. Conclusions and Future Work

Effective goal recognition holds considerable promise for player-adaptive games. Accurately recognizing players' goals enables digital games to proactively support gameplay experiences that feature nonlinear scenarios while preserving cohesion, coherence, and believability. This chapter has introduced a goal recognition framework based on Markov logic networks that accurately recognizes players' goals. Using model parameters learned from a corpus of player interactions in a complex, nonlinear game environment, the framework supports the automated acquisition of a goal recognition system that outperforms three baseline models.

There are several promising directions for future work on goal recognition models in open-ended digital games. First, the MLN model considered in this work deliberately leveraged minimal domain-specific knowledge. Devising systematic methods for encoding domain-specific information (e.g., interactive narrative structure, gameplay mechanics) is a promising direction for improving the predictive accuracy of the models. Second, investigating the efficiency and convergence rate of MLN goal recognizers is an important step for eventually embedding these models in runtime settings, because the real-time performance requirements of digital games demand highly efficient algorithms. Finally, integrating MLN goal recognition models into the CRYSTAL ISLAND game-based learning environment is a key objective of this work. The models would drive narrative-centered tutorial planners that dynamically adapt game events to enhance students' learning, problem solving, and engagement outcomes. This integration will enable empirical studies involving human subjects to examine the real-world impacts of goal recognition models and player-adaptive games.

Acknowledgments

The authors wish to thank members of the IntelliMedia Group of North Carolina State University for their assistance, Omer Sturlovich and Pavel Turzo for use of their 3D model libraries, and Valve Software for access to the Source^{TM} engine and SDK. This research was supported by the National Science Foundation under Grants REC-0632450, DRL-0822200, and IIS-0812291. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional support was provided by the Bill and Melinda Gates Foundation, the William and Flora Hewlett Foundation, and EDUCAUSE.

References

Baker, R. S., Corbett, A. T., Koedinger, K. R., and Wagner, A. Z. 2004. Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". In *Proceedings of the ACM Conference on Human Factors in Computing Systems,* 383-390. New York, NY: ACM.

Bethesda Softworks. 2011. *The Elder Scrolls V: Skyrim*. Retrieved from http://www.elderscrolls.com/skyrim.

Blaylock, N. and Allen, J. 2003. Corpus-based, Statistical Goal Recognition. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 1303-1308. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Bui, H. H. 2003. A General Model for Online Probabilistic Plan Recognition. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 1309-1315. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Carberry, S. 2001. Techniques for Plan Recognition. *User Modeling and User-Adapted Interaction*, 11(1-2): 31-48.

Charniak, E. and Goldman, R. P. 1993. A Bayesian Model of Plan Recognition. *Artificial Intelligence*, 64(1): 53-79.

Chu, Y., Song, Y. C., Levinson, R., and Kautz, H. 2012. Interactive Activity Recognition and Prompting to Assist People with Cognitive Disabilities. *Journal of Ambient Intelligence and Smart Environments*.

Conati, C., Gertner, A., and VanLehn, K. 2002. Using Bayesian Networks to Manage Uncertainty in Student Modeling. *Journal of User Modeling and User-Adapted Interaction*, 12(4): 371-417.

Crammer, K. and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3: 951-991.

Geib, C. W. and Goldman, R. P. 2009. A Probabilistic Plan Recognition Algorithm Based on Plan Tree Grammars. *Artificial Intelligence*, 173(11): 1101-1132.

Geib, C. and Goldman, R. 2011. Recognizing Plans with Loops in a Lexicalized Grammar. In *Proceedings of the Twenty-Fifth National Conference on Artificial Intelligence*, 958-963, Menlo Park, CA: AAAI Press.

Getoor, L. and Taskar, B. 2007. Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). Cambridge, MS: The MIT Press.

Gold, K. 2010. Training Goal Recognition Online from Low-Level Inputs in an Action-Adventure Game. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 21-26. Menlo Park, CA: AAAI Press.

Hoogs, A. and Perera, A. 2008. Video Activity Recognition in the Real World. In *Proceedings of the Twenty-Third National Conference on Artificial Intelligence*, 1551-1554. Menlo Park, CA: AAAI Press.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Koos, R. (1998). The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 256-265. San Francisco, CA: Morgan Kaufmann.

Hu, D. H. and Yang, Q. 2008. CIGAR: Concurrent and Interleaving Goal and Activity Recognition. In *Proceedings of the Twenty-Third National Conference on Artificial Intelligence*, 1363-1368. Menlo Park, CA: AAAI Press.

Kabanza, F., Bellefeuille, P. and Bisson, F. 2010. Opponent Behavior Recognition for Real-Time Strategy Games. In *Proceedings of the AAAI-10 Workshop on Plan, Activity, and Intent Recognition*. Atlanta, GA.

Kate, R. J. and Mooney, R. J. 2009. Probabilistic Abduction using Markov Logic Networks. In *Proceedings of the IJCAI-09 Workshop on Plan, Activity, and Intent Recognition*. Pasadena, CA.

Kautz, H. and Allen, J. F. 1986. Generalized Plan Recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 32-37. Menlo Park, CA: Morgan Kaufmann. An Offline Planning Approach to Game Plotline Adaptation. In *Proceedings of the 6th Artificial Intelligence and Interactive Digital Entertainment Conference*, 45-50. Menlo Park, CA: AAAI Press.

Laviers, K. and Sukthankar, G. 2011. A Real-Time Opponent Modeling System for Rush Football. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence,* 2476-2481. Menlo Park, CA: AAAI Press.

Lee, S. Y., Mott, B. W., and Lester, J. C. 2012. Real-Time Narrative-Centered Tutorial Planning for Story-Based Learning. In *Proceedings of the 11th International Conference on Intelligent Tutoring Systems,* 476–481. Berlin, Heidelberg: Springer.

Marcu, D. 2000. The Rhetorical Parsing of Unrestricted Texts: A Surface-Based Approach. *Computational Linguistics*, 26(3): 395-448.

Mojang. 2009. *Minecraft*. Retrieved from http://www.minecraft.net/.

Mott, B., Lee, S., and Lester, J. 2006. Probabilistic Goal Recognition in Interactive Narrative Environments. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 187-192. Menlo Park, CA.: AAAI Press.

Orkin and Roy 2007. The Restaurant Game: Learning Social Behavior and Language from Thousands of Players Online. *Journal of Game Developmen*t, 3(1): 39-60.

Pietra, S. D., P, Pietra, V. D., and Lafferty, J. 1997. Inducing Features of Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4): 380-393.

Pynadath, D. V. and Wellman, M. P. 2000. Probabilistic State-Dependent Grammars for Plan Recognition. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 507-514. Stanford, CA: Morgan Kaufmann Publishers Inc.

Rabiner, L. R.1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77 (2): 257-286.

Ramirez, M. and Geffner, H. 2010. Probabilistic Plan Recognition using off-the-shelf Classical Planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1121-1126. Menlo Park, CA: AAAI Press.

Richardson, M. and Domingos, P. 2006. Markov Logic Networks. *Journal of Machine Learning,* 62(1-2): 107-136.

Riedl, M. O., Saretto, C. J., and Young, R. M. 2003. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems*, 741–748. New York, NY: ACM.

Riedel, S. 2008. Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. In *Proceedings of the Twenty-Fourth Annual Conference on Uncertainty in Artificial Intelligence*, 468-475. Helsinki, Finland: AUAI Press.

Robison, J., McQuiggan, S. and Lester, J. 2009. Evaluating the Consequences of Affective Feedback in Intelligent Tutoring Systems. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction,* 37-42. IEEE.

Rowe, J. and Lester, J. 2010. Modeling User Knowledge with Dynamic Bayesian Networks in Interactive Narrative Environments. In *Proceedings of the Sixth International Conference on Artificial Intelligence in Interactive Digital Entertainment,* 57-62. Menlo Park, CA: AAAI Press.

Rowe, J., Shores, L., Mott, B., and Lester, J. 2011. Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *International Journal of Artificial Intelligence in Education*, 21(1-2): 115-133.

Sadilek, A. and Kautz, H. 2010. Recognizing Multi-Agent Activities from GPS Data. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1134-1139. Menlo Park, CA: AAAI Press.

Schmidt, C., Sridharan, N., and Goodson, J. 1978. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11, 45-83.

Singla, P. and Mooney, R. 2011. Abductive Markov Logic for Plan Recognition. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 1069-1075. Menlo Park, CA: AAAI Press.

Squire, K. 2008. Open-ended video games: A model for developing learning for the interactive age. In K. Salen (Ed.), *The Ecology of Games: Connecting Youth, Games, and Learning* (pp. 167-198). Cambridge, MA: The MIT Press.

Sturtevant, N. 2011. A Sparse Grid Representation for Dynamic Three-Dimensional Worlds. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 73-78. Menlo Park, CA: AAAI Press.

Yannakakis, G. 2012. Game AI Revisited. In *Proceedings of ACM Computing Frontier Conference.*

Zoeller, G. 2010. Game Development Telemetry. In *Proceedings of the Game Developers Conference*.

Author Information

Name: Eun Young Ha
Title: Postdoctoral Research Scholar
Institution: Department of Computer Science, North Carolina State University
Address: 890 Oval Drive, Raleigh, NC 27695-8206
Email: eha@ncsu.edu
Phone: 919-515-7534

Name: Jonathan P. Rowe
Title: Doctoral Candidate
Institution: Department of Computer Science, North Carolina State University
Address: 890 Oval Drive, Raleigh, NC 27695-8206
Email: jprowe@ncsu.edu
Phone: 919-515-7534

Name: Bradford W. Mott
Title: Senior Research Scientist
Institution: Department of Computer Science, North Carolina State University
Address: 890 Oval Drive, Raleigh, NC 27695-8206
Email: bwmott@ncsu.edu
Phone: 919-515-7534

Name: James C. Lester
Title: Distinguished Professor
Institution: Department of Computer Science, North Carolina State University
Address: 890 Oval Drive, Raleigh, NC 27695-8206
Email: lester@ncsu.edu
Phone: 919-515-7534