Deep LSTM-based Goal Recognition Models for Open-World Digital Games

Wookhee Min, Bradford Mott, Jonathan Rowe, James Lester
Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
{wmin, bwmott, jprowe, lester}@ncsu.edu

Abstract

Player goal recognition in digital games offers the promise of enabling games to dynamically customize player experience. Goal recognition aims to recognize players' high-level intentions using a computational model trained on a player behavior corpus. A significant challenge is posed by devising reliable goal recognition models with a behavior corpus characterized by highly idiosyncratic player actions. In this paper, we introduce deep LSTM-based goal recognition models that handle the inherent uncertainty stemming from noisy, non-optimal player behaviors. Empirical evaluation indicates that deep LSTMs outperform competitive baselines including single-layer LSTMs, *n*-gram encoded feedforward neural networks, and Markov logic networks for a goal recognition corpus collected from an open-world educational game. In addition to metric-based goal recognition model evaluation, we investigate a visualization technique to show a dynamic goal recognition model's performance over the course of a player's goal-seeking behavior. Deep LSTMs, which are capable of both sequentially and hierarchically extracting salient features of player behaviors, show significant promise as a goal recognition approach for open-world digital games.

Introduction

Human intelligence plays a pivotal role in interpersonal behaviors, communication, and relationships (Baker, Saxe, and Tenenbaum 2009; Sukthankar et al. 2014). Humans reason about others' cognitive and affective states and take actions according to the inferred states as well as the context and situation. A broad range of research has been undertaken to emulate social intelligence using artificial intelligence in the context of plan, activity, and intent recognition (PAIR).

Prior work on PAIR has largely focused on observation sequences in which an agent's actions are directly driven by concrete goals held by the agent. Observations may be influenced by several sources of uncertainty, such as noisy sensors and actions with stochastic outcomes. For instance, PAIR has been investigated in testbed applications such as home ambient intelligence (Sadri 2012), cyber security (Geib and Goldman 2009), terrorism detection (Jarvis, Lunt, and Myers 2005), intelligent interface agents (Armentano and Amandi 2011), and intelligent tutoring systems (Alvarez et al. 2015; Lee, Liu, and Popović 2014).

As an important line of PAIR research, goal recognition in digital games has been the subject of growing attention (Ha et al. 2011; Kabanza et al. 2013). Player goal recognition is the process of dynamically identifying the high-level objective that a player is attempting to achieve based on observable gameplay behaviors and states in the game world. Goal recognition offers the potential to dynamically adapt gameplay to player intentions, particular in interactive narrative (Riedl and Bulitko, 2013), game balancing (Lopes and Bidarra, 2011), procedural content generation (Shaker, Togelius, and Nelson 2015), and adaptive pedagogical planning in educational games (Ha et al. 2011; Mott, Lee, and Lester 2006).

Open-world digital games pose a significant challenge for goal recognition (Min et al. 2016a). These games do not explicitly present a set of goals to achieve, and there are a vast number of sub-optimal plans with which players can achieve their goals. In situations where players have limited prior experience with a digital game (e.g., serious games for training and education), it may be the case that players explore the game world rather than deliberately plan actions in order to achieve a specific gameplay objective. It is also possible that players will unintentionally achieve goals during exploration, suddenly abandon goals, or adopt new goals based upon prior events. These characteristics of open-world digital games produce highly idiosyncratic action sequences, and the task of recognizing players' goals exhibits significant uncertainty. Thus, devising reliable computational models is key to the success of goal recognition in these environments.

In our previous work (Min et al. 2016a, 2016b), we found that single-layer long short-term memory network (LSTM)-based goal recognition models harnessing distributed action representations significantly outperform *n*-gram encoded feedforward neural networks (FFNNs) pre-trained with stacked denoising autoencoders and Markov logic networks (MLNs), with respect to predictive accuracy, standardized convergence point, and *n*-early convergence rate in benchmark datasets. In this work, we further investigate LSTMs by extending shallow LSTMs to deep LSTMs that feature deep structures both in time and layers. We investigate how the depth in layers contributes to goal recognition performance on noisy gameplay behaviors, particularly compared to previous state-of-the-art shallow LSTM models. Model evaluations are performed using GOALIE (Generalized Observable Action Learning for Intent Evaluation) (Min et al. 2016b), a multidimensional goal recognition evaluation framework. Furthermore, we extend GOALIE to include a visualization technique as a supplementary tool to illustrate goal recognition models' dynamic predictive performance.

Related Work

Plan recognition can be formulated as a generalized task of goal recognition since plan recognition focuses on inferring plans and goals of observed agents (Sukthankar et al. 2014). While much previous plan recognition work has utilized a hand-crafted plan library (Geib and Goldman 2009), some other work has addressed plan recognition by learning a plan library in a data-driven approach (Vattam and Aha 2015) or inverse planning that removes the need for a plan library (Baker et al. 2009; Ramírez and Geffner 2011). However, these plan recognition approaches are not readily applicable to open-world digital games. For example, unlike the inverse planning work (Baker et al. 2009), which assumes a rational (or approximately rational) agent, players who have no prior experience with a game are likely to carry out exploratory, non-optimal actions, especially in early phases of gameplay. Goal recognition provides a viable solution for inferring players' intentions in open-world games. Previous work on goal recognition has shown that sequence or statistical relational models can efficiently and accurately recognize goals within highly noisy, idiosyncratic sequences of player behaviors (Min 2016).

There is a growing body of goal recognition work on handling noisy observation data due to an agent's mistakes or deceptive intention. Keren and colleagues (2015) extended *goal recognition design*, the task of which focuses on modifying an environment by limiting the set of available actions, to maximize the goal recognizer's early prediction capacity, particularly for non-optimal agents that have a budget for diverting from an optimal path. There is a notable distinction between Keren et al.'s work and ours: (1) our work focuses on devising robust goal recognition models with high predictive accuracy and early prediction capacity within a fixed virtual environment, and (2) a bounded nature to the agent's optimality is not strictly observed in our environment, since it is possible for a player to inadvertently achieve a goal during exploration. Sohrabi et al. (2016) investigated plan recognition as a process of inverse planning in a domain with unreliable observations. Their work specifically focused on either extra or partial observations, whereas our environment is fully observable but observations are inherently sub-optimal for achieving goals, rather than simply having extra actions on an optimal sequence.

Digital games can serve as laboratories for investigating computational techniques for goal and plan recognition. Bisson et al. (2015) examined recursive neural network-based decision models on a real-time strategy game (RTS) corpus. Recursive neural networks, which are a family of deep learning models first proposed for natural language parsing, have been investigated to automatically extract features that discriminate between correct and incorrect plan hypotheses and predict the plan hypothesis that best explains the observed action sequence. Evaluations suggest that the recursive neural network-based plan recognition approach outperforms two competitive baseline algorithms: a probabilistic plan-library based approach and an inverse planning approach. Kabanza et al. (2013) presented a heuristic weighted model counting algorithm that enables recognition of upper and lower bounds of posterior probabilities of goals in a RTS, and Synnaeve and Bessière (2011) investigated a probabilistic goal recognition approach in a RTS, in which plans are directly learned from game replays via unsupervised learning, while probabilistic decision models handle partial observations from agents.

In parallel to deep learning's significant advances in computer vision, speech recognition, and natural language processing (LeCun, Bengio, and Hinton 2015), deep learning has yielded considerable progress in goal and plan recognition (Bisson, Larochelle, and Kabanza 2015; Min et al. 2014; Min et al. 2016a). Min and colleagues (2014) investigated FFNNs pre-trained with stacked denoising autoencoders for goal recognition, which significantly outperformed previous state-of-the-art models based on MLNs. Compared to MLN-based approaches, which use a combination of hand-authored logic formulae and machine-learned weights, the deep learning approach eliminates labor-intensive feature engineering efforts by utilizing multi-level feature abstraction techniques. In more recent work by Min and colleagues (2016a), they examined goal recognition with single layer LSTMs leveraging distributed action representations. The results show that the LSTM-based goal recognition approach achieved state-of-the-art predictive accuracy for an open-world educational game goal recognition corpus.

Figure 1: CRYSTAL ISLAND open-world educational game.

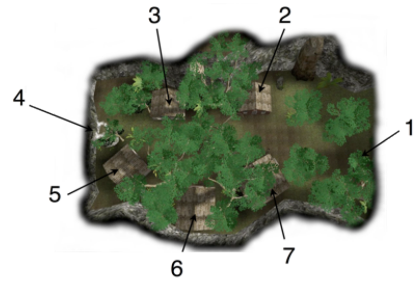

Figure 2: Map of the CRYSTAL ISLAND research camp.

CRYSTAL ISLAND Educational Game

CRYSTAL ISLAND (Figure 1) is a rich, virtual 3D educational game implemented using the Source game engine from Valve Software. In the game, players learn microbiology concepts, aligned with the North Carolina Standard Course of Study for eighth-grade microbiology, through an interactive science narrative. CRYSTAL ISLAND has been the subject of extensive empirical investigation, and has been found to provide substantial learning and motivational benefits (Rowe et al. 2011), while also offering significant challenge with fewer than half of players solving the mystery in less than an hour.

CRYSTAL ISLAND 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 a remote island. Players assume the role of a visiting investigator to the island, who is drawn into a mission to save the research team from the outbreak. Players interact with CRYSTAL ISLAND from a first-person viewpoint, using a diverse set of actions occurring in seven major locations of the research camp (Figure 2). For example, players move around the camp, pick up and drop objects, converse with virtual characters, view microbiology-themed posters and books to learn about infectious agents, take notes, use lab equipment to perform hypothesis testing, and complete a diagnosis worksheet. Players record their findings, hypotheses, and a final diagnosis in the diagnosis worksheet, and solve the mystery by submitting the correct worksheet to the camp nurse.

All player actions are logged by the CRYSTAL ISLAND game and stored for future data analyses. The data to be used for creating goal recognition models in this work was collected from a study involving 153 eighth grade students, aged 12–15 (M=13.3, SD=0.48) in a North Carolina public middle school. We removed 16 players from analysis due to incomplete data or prior experience with CRYSTAL ISLAND, and thus 137 players (males: 77, female: 60) are used to evaluate the goal recognition framework (Baikadi 2014).

CRYSTAL ISLAND Goal Recognition Corpus

The goal recognition work assumes that a given sequence of actions maps to a single goal, and no interleaving occurs between actions associated with different goals. During the data collection, we did not directly observe players' goals as they played CRYSTAL ISLAND. Rather, players' goals were identified during a post-hoc analysis, in which we labeled all actions between the previously achieved goal and the current goal with the label of the current goal that the player will next achieve (Min et al. 2014).

Under these conditions, goal recognition is cast as a multiclass classification problem in which a goal recognition model predicts the most likely goal associated with the currently observed sequence of actions after the previously observed goal. The CRYSTAL ISLAND goal recognition corpus includes 137 players' gameplay data that consists of 77,182 player actions (i.e., the total number of possible goal recognitions) and 893 achieved goals, with an average of 86.4 player actions per goal.

A key step that must precede model-training is to encode data in a format that machine-learning algorithms can take as input (i.e., features) and output (i.e., labels). Player actions are encoded with four properties: action type, location, narrative state, and previously achieved goals (Min et al. 2014).

- **Action Type**: CRYSTAL ISLAND includes 19 distinct types of player actions (e.g., talk, move, pick up).
- **Location**: CRYSTAL ISLAND includes 39 fine-grained and non-overlapping sub-locations that decompose the seven major camp locations (Figure 2).
- **Narrative State**: Four milestone narrative events include discussing the illness with the nurse, testing the contaminated object, submitting a diagnosis to the nurse, and submitting a correct diagnosis to the nurse.
- **Previously Achieved Goals**: Eight previously achieved goals are available, including 'None' in case the player has not yet achieved any goals.

Players advance through CRYSTAL ISLAND's non-linear narrative by completing a set of goals, which are not directly presented to the players. In this work, seven goals are considered (Table 1). Each goal represents a key high-level objective collectively required to solve the final mystery of the game. Descriptive statistics of the distribution of the seven goals are shown in Table 1, which is based on the number of required actions required to achieve a goal. The majority class-based accuracy rate is 26.6%.

Once a goal is achieved, any future occurrence of the goal-achieving action is considered a normal action rather than a goal because players already know how to achieve the goal and might have performed the action as a step to achieve a new goal.

Goal	Distribution
Running lab 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 1: CRYSTAL ISLAND: Distributions of goals.

Goal Recognition in CRYSTAL ISLAND

Due to the exploratory nature of player behavior in open-world digital games, goal recognition models should robustly handle cyclical relationships between player goals and actions (Ha et al. 2011; Min et al. 2016a). Players' previously achieved goals may inform their subsequent actions, and their current actions may influence their upcoming goals.

These characteristics of open-world digital games have inspired the investigation of a set of machine learning techniques for goal recognition. Ha and colleagues (2011) investigated Markov logic networks that are well suited for machine learning tasks in domains with complex associations between modeled entities such as actions and goals in goal recognition. Min et al. (2014) examined *n*-gram encoded feedforward neural networks, where previous actions and previously achieved goals are encoded in the feature set and are jointly utilized to predict the player's current goal.

In contrast to these two methods that formulate the sequential and cyclical relationships between actions and goals using a fixed length of inputs and outputs, sequence labeling techniques can take variable length inputs without constraining them to a fixed size. This flexible modeling capacity provided by sequence labeling techniques is well suited for capturing sequential, complex patterns across players' previous behavior, achieved goals, and current goal. This advantage has motivated the investigation of goal recognition models (Min et al. 2016a) based on long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber 1997). In this work, we investigate deep LSTMs featuring stacked LSTM layers.

LSTM Background

LSTMs are a variant of recurrent neural networks (RNNs) that are specifically designed for sequence labeling of temporal data. LSTMs have achieved high predictive performance in various sequence labeling tasks, often outperforming standard recurrent neural networks by leveraging a longer-term memory than standard RNNs and effectively addressing the vanishing gradient problem (LeCun, Bengio, and Hinton 2015).

LSTMs feature a sequence of memory blocks that include one or more self-connected memory cells (c_t) along with three gating units: an input gate (i_t), a forget gate (f_t), and an output gate (o_t). In LSTMs, the input and output gates modulate the incoming signals ($\widetilde{c_t}$, the candidate value for the memory cell state) and outgoing signals (c_t) to the memory cell, respectively, and the forget gate controls whether the previous state of the memory cell (c_{t-1}) is remembered or forgotten, where t denotes a time step.

Deep LSTMs (Figure 3A) extend shallow LSTMs by stacking multiple layers of LSTMs on top of each other (Graves, Mohamed, and Hinton 2013). Compared to the shallow LSTMs, deep LSTMs capture multi-level hierarchical representations within a time step, while preserving long term dependencies in the input sequence across time. The memory cell output (h_t^{n-1}) in the layer n-1 at time t is fed as input to the LSTM in the upper layer n at time t, while the lowest LSTM layer takes as input the original input sequence (x) as in shallow LSTMs, as illustrated in Figure 3A. Equations (1–6) present how deep LSTMs operate in the nth layer at time t. As noted, in deep LSTMs, the W matrices transform the output in the layer below (h_t^{n-1}), and, only for the lowest LSTM layer (in case of n=1), h_t^0 corresponds to x_t, which is the original input at time t.

$$i_t^n = \sigma(W_i^n h_t^{n-1} + U_i^n h_{t-1}^n + b_i^n) \tag{1}$$

$$f_t^n = \sigma(W_f^n h_t^{n-1} + U_f^n h_{t-1}^n + b_f^n) \tag{2}$$

$$\widetilde{c_t^n} = tanh(W_c^n h_t^{n-1} + U_c^n h_{t-1}^n + b_c^n) \tag{3}$$

$$c_t^n = i_t^n \widetilde{c_t^n} + f_t^n c_{t-1}^n \tag{4}$$

$$o_t^n = \sigma(W_o^n h_t^{n-1} + U_o^n h_{t-1}^n + b_o^n) \tag{5}$$

$$h_t^n = o_t^n \, tanh(c_t^n) \tag{6}$$

Deep LSTMs have been successfully examined in acoustic modeling such as speech recognition (Graves, Mohamed, and Hinton 2013). The enhanced expressiveness allowed by multiple layers in the LSTM network structure, along with the successful demonstration of shallow LSTM models for goal recognition (Min et al. 2016a), inspires our work to investigate deep LSTMs, which is expected to extract complex, sequential, hierarchical patterns underlying noisy gameplay behaviors in the process of achieving goals.

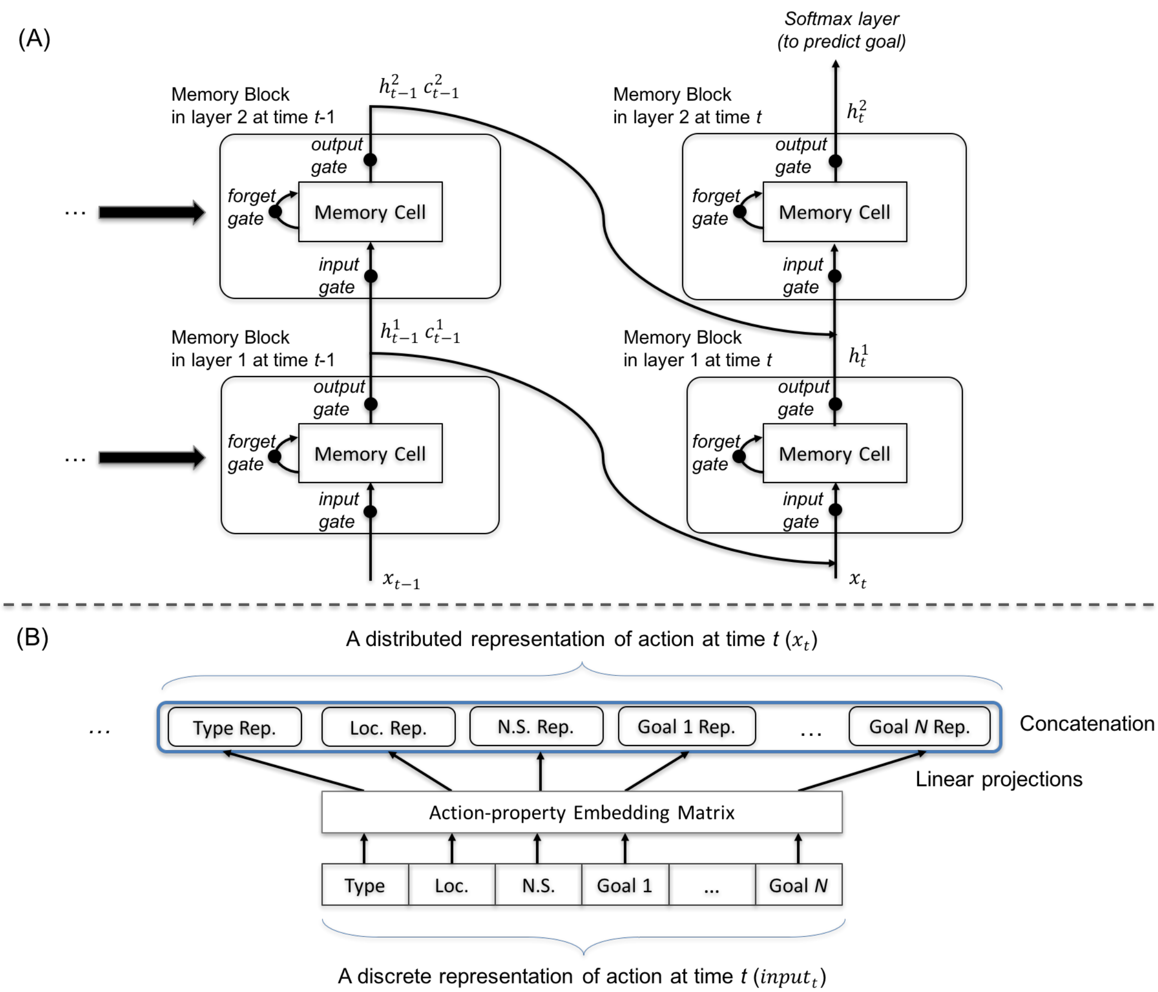

Figure 3: (A) Deep LSTMs (Graves, Mohamed, and Hinton 2013) with two LSTM layers for goal recognition. The cell output vector, h_t^2, at the last time step (t) in the highest layer (2) is utilized to predict the goal associated with x_t using a softmax layer. (B) A distributed action representation (x_t) generated from a discrete action representation ($input_t$) (Min et al. 2016a). An action-property embedding matrix linearly maps each action property (i.e., action type, action location, narrative state, previously achieved goals) in a discrete vector space onto a continuous vector space, and all N+3 continuous vectors are concatenated to generate a single distributed action representation, x_t (N=7 in this work). The induced x_t is fed into the LSTM memory block at time t.

Deep LSTM-Based Goal Recognition

To represent an action input in deep LSTMs, we follow the distributed action representation approach introduced in (Min et al. 2016a). Distributed vector representations were initially investigated on textual data for language modeling (Bengio et al. 2003). Distributed representations of words have been successfully examined in a wide range of natural language processing tasks, such as language parsing (Socher, Manning, and Ng 2010) and sentiment analysis (Le and Mikolov 2014). Distributed action representation-based LSTMs were also shown to outperform discrete action representation-based LSTMs for goal recognition, with respect to predictive accuracy, standardized convergence point, and N-early convergence rate (Min 2016).

In order to train distributed action embeddings, we utilize a 10-dimensional discrete vector as input for the deep LSTM (Figure 3B). The first three dimensions of the vector are allocated to represent the action type, action location, and current narrative state with integer-based indices, while the following seven dimensions represent a sequence of previously achieved goals (seven goals in total in CRYSTAL ISLAND) also with integer-based indices.

Distributed action embeddings are managed in an action-property embedding matrix that is randomly initialized following a uniform distribution (max: 0.05, min: -0.05) and is fine-tuned using supervised machine learning. Since an action consists of multiple properties (e.g., type, location, narrative state), distributed action representations are managed on a per-property basis. To operationalize this, a comprehensive action-property embedding matrix, the size of which is

74 (the total number of possible values of action properties computed as 19+39+8+8) by d (embedding size), is created. As the action-property embedding matrix operates as a linear transformation, the entire network is end-to-end trainable using a backpropagation method via supervised learning.

Each action in the sequence is encoded with a single distributed representation by concatenating the ten action property-based representations that constitute the action. The process for creating this distributed action representation is shown in the top layer in Figure 3B. As each action property has a d-dimensional continuous vector space, the size of a concatenated distributed action embedding to represent an action, x_t, is (10 x d).

At recognition time, a sequence of actions is sequentially fed into the LSTM model in the recurrent neural network formalism. The final memory cell output vector (e.g., h_t^2 in Figure 3A) is used to predict the most likely goal for the sequence of actions in a softmax layer, which is interpreted as a calculation of posterior probabilities of goals.

Evaluation

We evaluate goal recognition models' performance on the CRYSTAL ISLAND data corpus using the GOALIE framework. GOALIE (Min et al. 2016b) is a multidimensional evaluation framework for player goal recognition, equipped with a set of metrics that measure goal recognition models' performance from various perspectives, such as predictive accuracy and early prediction capacity. Among evaluation metrics, Min et al. (2016b) pointed out that the conventional convergence point metric can be misleading in representing models' early prediction capacity, since the metric ignores non-converged action sequences, in which the goal prediction on the last action is incorrect. To address this challenge, Min et al. suggested the *standardized convergence point* metric that takes into consideration all action sequences penalizing non-converged sequences (2016b). Thus, we report the goal recognition performance in terms of the accuracy rate, N-early convergence rate, and standardized convergence point. The N-early convergence rate measures the percentage of action sequences in which the last (N+1) goal predictions are correct. Standardized convergence point measures how quickly a goal recognition model can consistently produce accurate predictions. Lower is better for the standardized convergence point, while, for all other metrics, higher is better.

We evaluate four computational goal recognition approaches: Markov logic networks (MLN) (Baikadi 2014), n-gram encoded feedforward neural networks pre-trained using stacked denoising autoencoders (FFNN) (Min et al. 2014), shallow LSTMs with one single LSTM layer (LSTM-1) (Min et al. 2016a), and deep LSTMs with N stacked LSTM layers (LSTM-N), in which N is greater than one. The four computational approaches are evaluated using 10-fold cross-validation, where we use the same player-level data split for fair comparisons across the approaches.

For FFNN, LSTM-1, and LSTM-N, a configuration of model hyperparameters is identified using an automated grid search, and only the best performing model with respect to cross-validation accuracy rate is reported. MLNs, on the other hand, utilized human expert-crafted logic formulae in terms of discovery events, domain-specific representations of user progress specifically targeted to the digital game (Baikadi 2014). For LSTM-N, we explore two hyperparameters: N among {2, 3} and the number of hidden units among {25, 50, 100}, which are consistently used for all LSTM layers. For other hyperparameters, we set the dropout rate (Srivastava et al. 2014) and action embedding size to 0.75 and 20, respectively. Further, we use a softmax layer for classifying given sequences of actions, adopt a mini-batch gradient descent with the mini-batch size of 128, and utilize categorical cross entropy for the loss function and the Adam stochastic optimizer (Kingma and Ba 2014). For training efficiency, action sequences greater than ten are pruned to keep only the last ten actions. Finally, the training process stops early if the validation score has not improved within the last seven epochs. In this work, 10% of the training data is used to determine early stopping, while 90% is utilized for supervised training. The maximum number of epochs is set to 100.

	MLN	FFNN	LSTM-1	LSTM-2
Accuracy Rate (%)	55.21	62.43	66.35	**67.68**
Stand. Convergence point (%)	67.66	62.66	53.19	**48.69**
0E-Convergence Rate (%)	49.09	70.06	71.32	**72.91**
1E-Convergence Rate (%)	46.71	64.93	68.81	**71.90**

Table 2: Averaged rates of MLN, FFNN, LSTM with a single layer (LSTM-1), and LSTM with two layers (LSTM-2). *N*E-Convergence Rate denotes *N*-early convergence rate.

Table 2 presents results of the four computational goal recognition approaches. For deep LSTMs, LSTM-2 featuring two stacked LSTM layers each with 25 hidden units achieves the highest accuracy rate (67.68%) outperforming the best performing LSTM-3 networks (66.26%) also with 25 hidden units. LSTM-2 not only achieves the highest accuracy rate (i.e., greatest predictive performance), but also achieves the highest 0- and 1-early convergence rate and the lowest standardized convergence point, which together suggests that deep LSTMs are state-of-the-art for goal recognition in CRYSTAL ISLAND, as measured by the GOALIE evaluation framework.

We further investigate the performance of goal recognizers in CRYSTAL ISLAND using a visualization tool, which illustrates goal recognition models' dynamic performance over time. To achieve this, we extend the scope of the current

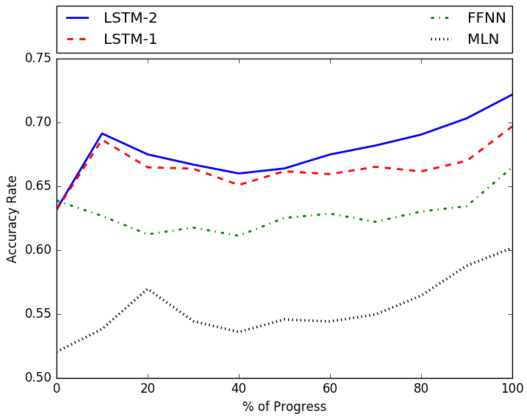

Figure 4: Average goal recognition accuracy over the course of the observation sequence.

GOALIE framework to include a graphical illustration in addition to its set of descriptive metrics.

The x-axis in Figure 4 denotes player progress in an action sequence, and the y-axis indicates the accuracy rate as the player progresses in the action sequence to achieving the goal. For example, suppose a player performs three actions A_1, A_2, and A_3 to achieve a goal, G_1, and a goal recognition model dynamically predicts the player's goal three times for each observed action, such as G_1 (correct) for A_1, G_2 (incorrect) for A_1 and A_2, and G_1 (correct) for A_1, A_2, and A_3. First, to visualize the dynamically changing performance of a goal recognition model, we create 11 bins (0–5%, 5–15%, 15–25%, …, 85–95%, 95–100%) to represent a player's progress in an action sequence. Except for the first and last bin, each bin has a range of 10%, while the first and last bins have a 5% range. These bins serve as the x-axis. Returning to the example, since the goal recognition model correctly predicts the goal for the first action (100% accuracy rate for progress of 33.3% toward the goal), we fill the first bin (0–5%) to the fourth bin (25–35%) with 100%. Then, for the second goal prediction, the accuracy rate is 0% as it predicts the goal wrong, when the current progress toward the goal is 66.7%. To add this information, we fill the fifth bins (35–45%) to the eighth bins (65–75%) with 0%. Finally, for the last action's goal prediction, we fill the ninth (75–85%) to eleventh bins (95–100%) with 100%, as the goal recognition model recognizes the correct goal. This process is repeated for all action sequences in the test sets (ten different test sets in 10-fold cross-validation), and all scores in each bin are averaged to have a single value. Note that the size of bins is not restricted to 11, but we use this value for an illustration.

It is not surprising that all the graphs have an increasing pattern in general, as goal recognizers can predict the player goals more accurately when presented more observed evidence. However, it is interesting to see that there is a peak around 20% of the progress for LSTM-2, LSTM-1 and MLN, after which the predictive accuracy continuously decreases until about 40%. Even though this pattern has not been rigorously scrutinized, we speculate that the large outdoor region that players pass through to achieve a goal situated in another building results in some confusion for the goal recognition models, while even the initial set of actions (before 20% of the progress) taken for achieving a goal provides stronger explanations (e.g., the player is more likely to achieve a proximal goal from the location where the last goal was achieved) for the models. The deep LSTM (LSTM-2) appears to perform better in dealing with these noisy interactions, consistently outperforming the shallow LSTM (LSTM-1) after the initial set of actions. For all the goal recognition approaches, the accuracy rate is highest at the end of the action sequences, which demonstrates the goal recognition models can more accurately model noisy player behaviors as the behavioral cues get more directly related to the current goal.

Conclusion

Player goal recognition is a core plan, activity, and intent recognition task for open-world digital games. We have introduced deep LSTM-based goal recognition models for an educational game corpus characterized by non-optimal player behaviors. Empirical evaluations with the GOALIE framework indicate that deep LSTMs with two stacked LSTM layers achieve the most reliable results across accuracy rate, standardized convergence point, and *N*-early convergence rate metrics. We additionally present a visualization technique as a supplementary tool for GOALIE. This graphical representation illustrates how goal recognition models dynamically operate, as players progress toward achieving various goals in the game.

In the future, it will be important to investigate other forms of recurrent neural networks along with different optimization and regularization techniques in order to identify better models. Also, there exist a broad range of opportunities to enhance the GOALIE evaluation framework by incorporating additional metrics and graphs that represent currently unmeasured aspects of goal recognition models' performance. With regard to the current visualization work, it will be interesting to understand players' behaviors along with predicted goals, and systematically examine patterns found in the graph. The graph introduced in this work aggregates all goal recognition results regardless of the goals, thereby providing a single graph. Generating a graph per goal can provide a more fine-grained analysis by illustrating how models operate differently based on goals. Moreover, it will be interesting to investigate how to illustrate dynamic goal recognition performance in terms of the scenario completion rather than the action sequence. For goal recognition in educational games, it will be important to investigate the relationships between the players' goals and learning outcomes. Finally, it will be important to investigate how goal recognition models operate at run-time to most effectively drive gameplay personalization in player-adaptive games.

References

Alvarez, N.; Sanchez-Ruiz, A.; Cavazza, M.; Shigematsu, M.; and Prendinger, H. 2015. Narrative Balance Management in an Intelligent Biosafety Training Application for Improving User Performance. *International Journal of Artificial Intelligence in Education* 25(1):35–59.

Armentano, M. G., and Amandi, A. A. 2011. Personalized detection of user intentions. *Knowledge-Based Systems* 24(8):1169–1180.

Baikadi, A. 2014. *Discovery-based Goal Recognition in Interactive Narrative Environments*. North Carolina State University.

Baker, C.; Saxe, R.; and Tenenbaum, J. 2009. Action Understanding as Inverse Planning. *Cognition* 113(3):329–349.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research* 3:1137–1155.

Bisson, F.; Larochelle, H.; and Kabanza, F. 2015. Using a Recursive Neural Network to Learn an Agent's Decision Model for Plan Recognition. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 918–924. AAAI Press.

Geib, C., and Goldman, R. 2009. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence* 173(11):1101–1132.

Graves, A.; Mohamed, A.; and Hinton, G. 2013. Speech Recognition With Deep Recurrent Neural Networks. In *IEEE international conference on acoustics, speech and signal processing*, 6645–6649.

Ha, E. Y.; Rowe, J. P.; Mott, B. W.; and Lester, J. C. 2011. Goal Recognition with Markov Logic Networks for Player-Adaptive Games. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 32–39.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1–32.

Jarvis, P. A.; Lunt, T. F.; and Myers, K. L. 2005. Identifying terrorist activity with AI plan-recognition technology. *AI Magazine* 26(3):73–81.

Kabanza, F.; Filion, J.; Benaskeur, A. R.; and Irandoust, H. 2013. Controlling the hypothesis space in probabilistic plan recognition. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 2306–2312. AAAI Press.

Keren, S.; Gal, A.; and Karpas, E. 2015. Goal Recognition Design for Non-Optimal Agents. In *Proceedings of the AAAI Conference*, 3298–3304.

Kingma, D. P., and Ba, J. L. 2015. Adam: a Method for Stochastic Optimization. In *Proceedings of International Conference on Learning Representations*.

Le, Q., and Mikolov, T. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of International Conference on Machine Learning,* 1188–1196.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep Learning. *Nature* 521(7553):436–444.

Lee, S. J.; Liu, Y.; and Popović, Z. 2014. Learning Individual Behavior in an Educational Game: A Data-Driven Approach. In *Proceedings of the 7th International Conference on Educational Data Mining*, 114–121.

Lopes, R., and Bidarra, R. 2011. Adaptivity challenges in games and simulations: A survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3(2):85–99.

Min, W. 2016. *Generalized Goal Recognition Framework for Open-World Digital Games*. North Carolina State University.

Min, W.; Ha, E. Y.; Rowe, J.; Mott, B.; and Lester, J. 2014. Deep Learning-Based Goal Recognition in Open-Ended Digital Games. In *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 37–43.

Min, W.; Mott, B.; Rowe, J.; Liu, B.; and Lester, J. 2016a. Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2590–2596.

Min, W.; Baikadi, A.; Mott, B.; Rowe, J.; Liu, B.; Ha, E. Y.; and Lester, J. 2016b. A Generalized Multidimensional Evaluation Framework for Player Goal Recognition. In *Proceedings of the 12th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 197–203.

Mott, B.; Lee, S.; and Lester, J. 2006. Probabilistic goal recognition in interactive narrative environments. In *Proceedings of the National Conference on Artificial Intelligence*, 187–192.

Ramírez, M., and Geffner, H. 2011. Goal recognition over POMDPs: Inferring the intention of a POMDP agent. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2009–2014.

Riedl, M. O., and Bulitko, V. 2013. Interactive Narrative: An Intelligent Systems Approach. *AI Magazine* 34(1):67–77.

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.

Sadri, F. 2012. Intention Recognition in Agents for Ambient Intelligence: Logic-Based Approaches. *Agents and Ambient Intelligence*, 197–240.

Shaker, N.; Togelius, J.; and Nelson, M. J. 2015. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Springer.

Socher, R.; Manning, C. D.; and Ng, A. Y. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, 1–9.

Sohrabi, S.; Riabov, A.; and Udrea, O. 2016. Plan recognition as planning revisited. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Sukthankar, G.; Geib, C.; Bui, H.; Pynadath, D.; and Goldman, R. P. 2014. *Plan, Activity, and Intent Recognition*. Elsevier.

Synnaeve, G., and Bessière, P. 2011. A Bayesian model for opening prediction in RTS games with application to StarCraft. In *Proceedings of IEEE Conference on Computational Intelligence and Games*, 281–288.

Vattam, S. S., and Aha, D. W. 2015. Case-Based Plan Recognition Under Imperfect Observability. In *Proceedings of the International Conference on Case-Based Reasoning*, 381–395. Springer.