Deep Learning-Based Goal Recognition in Open-Ended Digital Games

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

Abstract

While many open-ended digital games feature non-linear storylines and multiple solution paths, it is challenging for game developers to create effective game experiences in these settings due to the freedom given to the player. To address these challenges, goal recognition, a computational player-modeling task, has been investigated to enable digital games to dynamically predict players' goals. This paper presents a goal recognition framework based on stacked denoising autoencoders, a variant of deep learning. The learned goal recognition models, which are trained from a corpus of player interactions, not only offer improved performance, but also offer the substantial advantage of eliminating the need for labor-intensive feature engineering. An evaluation demonstrates that the deep learning-based goal recognition framework significantly outperforms the previous state-of-the-art goal recognition approach based on Markov logic networks.

Introduction

Recent years have seen a growing number of digital games featuring open-ended environments that provide players with significant autonomy over the goals they pursue and the plans they use to achieve their goals (Squire 2008). The non-linear goals, story plots, and multiple solution paths highlighted in these open-ended environments can promote player's engagement and support increased replayability. However, designing and developing these games pose significant challenges for game developers, who have to simultaneously provide cohesive narratives and storyworld events in a timely and relevant fashion while supporting dynamic progress in the game.

Previous research has addressed these challenges in the context of player-adaptive games, which dynamically adapt players' in-game experiences through computational modeling of player experience. Such player-adaptive games have incorporated computational techniques such as drama management (Li and Riedl 2010; Mateas and Stern 2005; Roberts, Cantino, and Isbell 2007; Thue et al. 2010), procedural content generation (Jennings-Teats, Smith, and Wardrip-Fruin 2010; Shaker, Yannakakis, and Togelius 2010), and adaptive pedagogical planning in educational games (Mott, Lee, and Lester 2006; Ha et al. 2011; Min et al. 2013).

As an approach to player modeling, goal recognition has drawn increasing attention in Game AI. Goal recognition, a restricted form of plan recognition, focuses on identifying the player's concrete objectives, given a series of low-level actions in virtual environments (Carberry 2001; Kautz and Allen 1986; Schmidt, Sridharan, and Goodson 1978). As a promising research area, goal recognition has been investigated in serious games, where it contributes to assessing players' learning progress, and thus assists in delivering tailored pedagogical strategies attuned to individual learners (Conati, Gertner, and VanLehn 2002; Johnson 2010; Lee, Mott, and Lester 2012). In addition to runtime adaptation of games, goal recognition can also support the game design process by facilitating interpretation of raw game logs along with identified goals, and providing insights to improve player experiences (Zoeller 2010).

Previous goal recognition work has traditionally focused on sequences of user actions derived from well-defined goals and plans (Blaylock and Allen 2003; Carberry 2001; Charniak and Goldman 1993; Geib and Goldman 2009; Hu and Yang 2008; Kautz and Allen 1986; Pynadath and Wellman 2000). In contrast to these environments, digital games with non-linear goals often do not explicitly present goals to players; rather, players identify and achieve goals in an exploratory fashion. In these situations, players' goals are hidden from the system's perspective, and thus they must be dynamically inferred based on observable features such as players' low-level actions and triggered events.

This paper proposes a computational framework for player goal recognition based on stacked denoising autoencoders, a *deep learning* technique. In this work, goal recognition is formalized as predicting the next sub-goal in a path to achieving the game's final objective. The goal recognition framework therefore identifies sub-goals based on the players' preceding sequence of actions. This framework has been evaluated with CRYSTAL ISLAND, a non-linear story-centric educational game with ill-defined goals. CRYSTAL ISLAND presents players with a single high-level objective, namely, to solve a science mystery. Findings from the evaluation suggests that the deep learning approach significantly outperforms the previous state-of-the-art approach based on Markov logic networks (MLN) in terms of prediction accuracy rates on the same dataset. In addition to the predictive performance improvement, deep learning's representation learning constitutes a significant step forward with respect to MLN-based goal recognition models because it eliminates the labor-intensive need to manually engineer hand-crafted features in the form of logic formulae.

Related Work

Goal recognition and plan recognition have been widely investigated in the context of digital games to support tasks such as determining players' objectives in action-adventure games and creating adaptable computer-controlled opponents. Kabanza, Bellefeuille, and Bisson (2010) explored challenges with behavior recognition in real-time strategy games to create adaptable computer-controlled opponents. Their work extended Geib and Goldman's (2009) PHATT algorithm to perform intent recognition on the opponents' behaviors. Gold (2010) investigated Input-Output Hidden Markov Models for recognizing high-level player goals in an action-adventure game, and the model was compared to a hand-authored finite state machine, a common computational framework used in commercial games.

Closely related to the work presented in this paper, Mott et al. (2006) examined several probabilistic goal recognition models to support dynamic narrative planning in an educational game. In a related game-based learning environment, Ha et al. (2011) used Markov logic networks (MLNs) to recognize players' goals from observed sequences of player actions to support personalized experiences (e.g., narratives and storyworld events) and targeted pedagogical planning. Our research is the first to propose a computational approach based on deep learning for goal recognition, and is evaluated against state-of-the-art MLN models (Ha et al. 2011).

Goal Recognition Corpus

To investigate the effectiveness of deep learning-based goal recognition models for open-ended digital game, data was collected from student interactions with the CRYSTAL ISLAND game-based learning environment (Figure 1). 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 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. 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.

Figure 1. CRYSTAL ISLAND virtual environment.

Players advance through CRYSTAL ISLAND's non-linear narratives by completing a partially ordered sequence of goals. In this work, seven goals are considered: *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.

Players interact with CRYSTAL ISLAND using a diverse set of actions occurring in the seven major locations of the research camp (Figure 2): a large *outdoors* region, an *infirmary*, a *living quarters*, a *waterfall*, the *lead scientist's*

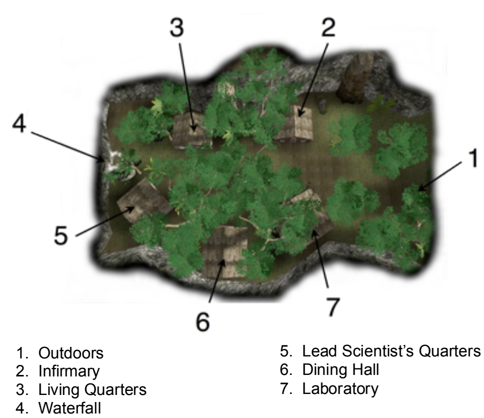

Figure 2. Map of the CRYSTAL ISLAND research camp.

quarters, a *dining hall*, and a *laboratory*. Players can perform actions that include: *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*.

All player actions are logged by the CRYSTAL ISLAND learning environment and stored for later analysis. The data used for creating the goal recognition models was collected from a study involving 137 eighth grade students from a public middle school.

Deep Learning-Based Goal Recognition

Similar to previous work on goal recognition (Blaylock and Allen 2003), we define goal recognition as the task of predicting the most likely goal for a given sequence of observed player actions in the environment. Following Ha et al. (2011), this work assumes that a given sequence of actions maps to a single goal, and no interleaving occurs between actions associated with different goals, since our existing dataset does not lend itself to this type of analysis. Under these conditions, goal recognition is cast as a multiclass classification problem in which a learned classifier predicts the most likely goal associated with the currently observed sequence of actions after the previously observed goal. In this work, a player action is encoded with five properties: *action type*, *location*, *narrative state*, *previously achieved goals*, and *action argument*. The *action argument* is a newly added property that was excluded in Ha et al.'s work due to data sparsity issues (Ha et al. 2011). The feature set including this additional property is called the *expanded feature set*, while the feature set that only considers the original four properties is called the *reduced feature set*. The effects of utilizing this extra property are explored in the evaluation section. The framework considers five properties of player actions:

- **Action Type**: The type of current action taken by the player, such as "*move*" to a particular location, and "*talk*" to a non-player character. Our data includes 19 distinct types of player actions.
- **Action Argument**: The argument taken by the action, such as talk to "*Robert*" and test "*milk*" using the laboratory's testing equipment. Our data, in sum, includes 90 distinct action arguments.
- **Location**: The location 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.
- **Narrative State**: An indication of the player's progress in solving the narrative scenario. Narrative state is represented as a vector of four binary variables. Each variable represents a milestone event within the narrative. The four milestone events 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*.
- **Previously Achieved Goals**: An indication of the previous goals achieved by the player. The previous *n-1* goals are considered for an *n*-gram encoded model. For example, the trigram model takes into consideration the two previously achieved goals, and the last three actions.

Goals represent key problem-solving steps in the CRYSTAL ISLAND science mystery. Two salient features of goals in the game environment influence encoding of the input data for our models. First, goals are dependent; some goals are completed in rapid succession to other goals, and a pair of goals is more likely to occur subsequently than other pairs. This is likely attributed to the geographical proximity in CRYSTAL ISLAND. Second, players are not provided explicit goals to achieve; rather, a cyclical causality between player actions and goals is possible, and players can learn about goals while interacting with the virtual environment. These correlations among goals and actions suggest that *n*-gram encoded models (especially when $n > 1$) might be able to improve the goal recognition predictive performance over unigram encoded models.

The current work employs *n*-gram augmented deep learning (DL) that pre-trains hierarchical representations through multi-layer abstraction of data, often in the context of artificial neural networks (Hinton, Osindero, and Teh 2006; Bengio et al. 2007). Advantages of DL over conventional supervised learning techniques include (1) representation learning without requiring labor-intensive feature engineering relying on domain-specific details, and (2) unsupervised pre-training that leverages unlabeled training instances. It has been empirically shown that pre-training helps find a region of parameter space that can reach a better local optimum in a non-convex optimization graph (Erhan et al. 2010; Palm 2012).

We utilize *stacked denoising autoencoders* (SDAEs), an implementation of DL in the artificial neural network paradigm (Vincent et al. 2008), to solve the goal recognition task. SDAEs, an extension of stacked autoencoders (Bengio et al. 2007), offer several benefits such that (1) unsupervised initialization of layers with denoising techniques helps capture robust hierarchical representations that effectively deal with noise in inputs, and (2) denoising techniques help eschew from situations where basic autoencoders learn trivial weights (i.e., identity matrix for weights between layers) (Vincent et al. 2008). Additionally, SDAEs provide a representational benefit: each neuron can denote a feature such as an action type and location at a certain time. To provide input data to the artificial neural network formalism for the expanded feature set, the raw game data is encoded into data instances that consist of 152 features (19 action types + 90 action arguments + 39 locations + 4 narrative states) for the input layer and 7 features (7 goals) for the output layer in a unigram encoded model with the expanded feature set. This encoding scheme allows our domain to be compactly modeled for SDAE, and is especially beneficial in both model training and inference, by significantly reducing the number of required computations.

Deep Learning Background

Deep learning (DL) emerged from work on artificial neural networks (ANNs). The back-propagation algorithm using a gradient descent method has been popularly utilized to train ANNs since the mid-1980s (Russell et al. 1995); however, this algorithm has been shown to not be scalable to deep ANNs with multiple layers, not only for its slow learning rate, but also because it often gets trapped in a poor local optima (Deng and Yu 2014). To address these challenges, Hinton et al. (2006) proposed a deep belief network based layer-wise unsupervised pre-training technique leveraging restricted Boltzmann machines (RBMs), undirected probabilistic graphical models constrained to form a bi-partite graph with two layers: a visible layer and a hidden layer. Another class of layer-wise unsupervised pre-training algorithms utilizing autoencoders (AEs) instead of RBMs was introduced with a similar purpose (Bengio et al. 2007; Vincent et al. 2008). In conjunction with various pre-training algorithms, DL can further improve the model by fine-tuning the initialized weights during back-propagation. Successful domains leveraging DL include computer vision, natural language processing, and information retrieval (Erhan et al. 2010).

An AE is interpreted as a nonlinear generalization of principal component analysis, where it encodes high-dimensional input data into a low-dimensional output data by applying a deterministic function (Hinton and Salakhutdinov 2006). As a variant of AEs, stacked autoencoders (SAEs) perform layer-wise representation learning, sequentially for all layers (Bengio et al. 2007). To pre-train a model, in the first layer, a SAE encodes an example input vector x in the visible layer into $h(x)$ in the hidden layer, using the current weight parameters W_1 and an activation function s (e.g., sigmoid) (Equation 1). Then, it decodes the encoded input $h(x)$ to $z(x)$ using W_2 and s (Equation 2), and updates the current weight parameters W_1 and W_2 so as to minimize the reconstruction error between the original input x and the decoded input $z(x)$, for example using stochastic gradient descent. In the equations, b_1 and b_2 are biases for W_1 and W_2, respectively.

$$h(x) = s(W_1x+b_1). \tag{1}$$
$$z(x) = s(W_2h(x)+b_2). \tag{2}$$

Once the first layer is trained, the hidden neuron vector $h(x)$ activated by Equation 1 on the input vector x now serves as a visible input vector for the next hidden layer, and this step is iteratively applied to all hidden layers. Once the last hidden layer is trained, it is connected to a supervised layer that consists of output neurons, and all the weight parameters in the deep architecture are fine-tuned using supervised learning (Bengio et al. 2007).

We utilize stacked denoising autoencoders (SDAEs), an extension of SAE, for goal recognition. SDAEs address a commonly observed challenge posed by SAEs, such that W_1 often converges to a trivial solution (i.e., identity matrix), especially when the number of variables in the hidden layer is equal to or greater than the number of variables in the visible layer (Vincent et al. 2008). Figure 3 shows a conceptual illustration of how the SDAE algorithm learns weights for a visible layer and a hidden layer. A training example x in the visible layer is corrupted (in this work, we set random neurons to 0) into a partially destroyed input x', then x' is deterministically mapped to $h(x')$ in the hidden layer using Equation 1, and weight parameters (W_1 and W_2) are updated to minimize the reconstruction error (L) between the uncorrupted input x and the decoded input $z(x')$, similar to the manner in which SAEs operate. As a result, SDAEs leverage a stochastic mapping of x for the autoencoders' layer-wise pre-training and thus effectively eschew W_1 from simply converging to the identity matrix (Vincent et al. 2008).

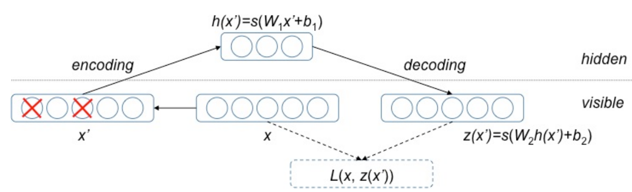

Figure 3. Conceptual illustration of layer-wise stacked denoising autoencoders; red crosses denote corruption.

Deep Learning for Goal Recognition

We utilize DeepLearnToolbox, a Matlab toolkit, to build a goal recognition classifier leveraging SDAE (Palm 2012). SDAE has an adjustable parameter set for pre-training; most of them are for the ANNs, while the corruption level (fraction of corrupted input neurons) is specific to the SDAE algorithm. Selecting an effective network topology (i.e., hyper-parameters) for ANNs often must be empirically determined, such as selecting a model with the minimum validation error among multiple variants of models (Svozil, Kvasnicka, and Pospichal 1997). Therefore, on one hand, we explore the model space by adjusting values of some parameters such as the input feature set (the expanded and reduced feature sets), the number of input neurons according to the n-gram encoding (792 at the most and 62 at the least), the number of hidden layers (2 or 3), and the number of epochs for autoencoders' gradient descent learning (1 or 2). On the other hand, we have fixed the following parameters: the number of neurons per a hidden layer (100), the gradient descent optimization method in back-propagation (stochastic gradient descent), the corruption level (0.5), the layer-wise learning rates, and the activation function (sigmoid). Specifically for the pre-training learning rates, we applied 1, 0.3, and 0.1 for the first, second, and third layer with respect to 3 hidden layer models, and 1 and 0.1 for the first and second layer with respect to 2 hidden layer models. For the supervised fine-tuning, 1 is used for every layer.

Evaluation

To evaluate the proposed SDAE-based goal recognition model, the data from the observation corpus is processed with the procedure introduced in Ha et al. (2011). First, all actions identified in the observation sequence that precede the current goal but follow the previous goal are labeled with the current goal. Second, actions that achieve goals are removed from the data, because it would be trivial to directly recognize goals from the goal-achieving actions. Finally, all actions taken after achievement of the last goal are ignored, since these player actions are more related to exploration of the game world solely for engagement, not resulting in any goals. The total number of goals achieved in the training data is 893, and the average number of player actions per goal is 86.4. Note that the 86.4 figure resulted from players' exploratory actions, and thus it does not necessarily represent the length of an efficient path to a goal. The most likely goal out of the seven is *running laboratory test on contaminated food* that has the probability of 26.6%.

Model evaluation is conducted along three dimensions: (1) comparison of accuracy rates between the current state-of-the-art MLN goal recognition models and the DL models, (2) impact analyses of adjustable parameters in DL, and (3) correlation tests among accuracy rates, *convergence rates*, and *convergence points* for DL models (Blaylock and Allen 2003). Convergence rate is a metric that measures the percentage of sequences that are eventually classified to the correct goal. Any sequence whose final action is predicted as belonging to the correct goal is said to have converged on the goal, and thus a higher number is better for this metric. Convergence point measures the percentage of a converged sequence that was observed before the correct goal was consistently predicted, and thus a lower number indicates improved performance.

First, accuracy rates are cross-compared among four models: a MLN based on the reduced feature set (MLN-R) that excludes action arguments, a MLN based on the expanded feature set (MLN-E) that includes action arguments, and two SDAE models (SDAE-R and SDAE-E) based on the two feature sets. Each model is trained and evaluated using 10-fold cross validation; in the cross validations, models use the same split of the training data for pairwise comparisons. Table 1 illustrates average accuracy rates, convergence rates, and convergence points of the four models, in which the SDAE models are the ones that achieve the highest accuracy rate from each feature set. Through empirical analyses, the most accurate model for SDAE-R is obtained by the five-gram setting with two hidden layers and 1 epoch, and the most accurate model for SDAE-E is achieved by the five-gram setting with two hidden layers and 2 epochs. The logic formulae used to generate the MLN models are described in Ha et al. (2011). They were hand-engineered, achieving sizable improvements over prior approaches.

Table 1. Averaged rates of MLN and DL.

	MLN-R	MLN-E	SDAE-R	SDAE-E
Accuracy Rates	48.4%	48.2%	**62.3%**	61.7%
Convergence Rates	43.5%	N/A	**70.9%**	66.9%
Convergence Points	32.1%	N/A	**42.8%**	44.3%

For pairwise comparisons of the models, we run the Friedman test, a non-parametric equivalent of repeated measures ANOVA, along with a post-hoc analysis with Wilcoxon signed-rank tests, since accuracy rates of folds do not necessarily follow normal distributions (Demšar 2006). Based on the Friedman test, there is a statistically significant difference in accuracy rates depending on the models, χ^2 (3) = 24.98, p < .001. The Wilcoxon signed-rank post-hoc tests indicate there are statistically significant improvements in accuracy rates for SDAE models over MLN models (all with Z = -2.8, p = .005), but MLN-R vs. MLN-E and SDAE-R vs. SDAE-E do not constitute a statistically significant difference.

Second, by aggregating fold-based validation accuracies per a distinct set of variant parameters, we evaluate the impact of DL parameters on the goal recognition predictive performances. The parameters are (1) n in n-gram input encoding: 1-5, (2) feature set types: expanded and reduced, (3) hidden layers: 2-3, and (4) epochs: 1-2. Pairwise comparisons are conducted on the validation result. Note that the n-gram test (5 groups) is conducted using the Friedman test geared with Wilcoxon signed-rank post-hoc tests, while the two other tests (2 groups each) are performed with Wilcoxon signed-rank tests (Demšar 2006). For n-gram, the Frieman test shows that the n-gram encoding elicit statistically significant differences (χ^2 (4) = 218.63, p < .001), and the post-hoc test indicates that every pair of models are different with statistical significance in terms of average accuracy rates (p < .001), other than four-gram vs. five-gram (Table 2). For the other statistical analyses, overall improvement in accuracy rates is found for the reduced feature set (Red.), 2 layer model (2 Lay.), and 1 epoch model (1 Ep.); however, there is no statistically significant difference in these tests (Table 3).

Lastly, convergence rate and convergence point view the goal recognition problem from different angles. Even though the goal recognition system consistently makes correct predictions on sequential actions (i.e., high accuracy rate), if it makes an incorrect prediction on the last action (i.e., low convergence rate), the model might be regarded as lacking robustness and reliability, considering the last action usually has more importance than priors. On the other hand, convergence point suggests how early the system can adapt to the player's current goal. A Spearman non-parametric correlation test is run to evaluate the correlation across these two metrics along with accuracy rate. The results indicate that there is a statistically significant positive correlation between accuracy rate and convergence rate (r = .69, p < .001), and a statistically significant negative correlation between accuracy rate and convergence point (r = -.70, p < .001), and convergence rate and convergence point (r = -.68, p < .001). To further examine the correlation, we ran two additional correlation tests on high performing models and low performing models, after separating the evaluation result into two groups based on the median of the models' accuracy rates. Out of these tests, it is noteworthy that the high performing models' accuracy rate is no longer in a strong correlation with the convergence point (r = -.02, p = .93).

Table 2. Averaged accuracy rates based on n-gram

Unigram	Bigram	Trigram	Four-gram	Five-gram
48.7%	51.5%	56.2%	60.0**%**	**60.4**%

Table 3. Averaged accuracy rates based on feature sets, the number of layers, and the number of epochs

Exp.	Red.	2 Lay.	3 Lay.	1 Ep.	2 Ep.
55.2%	**55.6%**	**55.7%**	55.0%	**55.4%**	55.3%

Discussion

Our DL based goal recognition model achieves a 28.7% marginal improvement over the state-of-the-art MLN model with a prediction rate accuracy of 62.3% (Table 1). A possible explanation for the DL-based model's strong performance compared to other machine learning techniques is that the DL's objective, "to discover high-level representations of raw low-level data," is inherently related to the latent intent of the goal recognition task, "to recognize higher-level patterns that result in goals using low-level action sequences."

The evaluation results for different parameter settings suggest that previous actions along with achieved goals serve as strong predictors to improve the goal recognition performance (Table 2). Considering the highest performance is obtained using the five-gram encoding, it might be worth investigating higher *n*-gram encodings in the future. Other parameters such as the number of layers and the number of epochs do not show any statistically significant difference under our constrained experiments. Interestingly, the expanded feature set that considers action arguments turned out not to improve the model's predictive performance as echoed in the MLN experiments, which indicate the action arguments might be too noisy or sparse to be served as strong predictors in DL (Table 3).

It is straightforward to see such a strong correlation among accuracy rate, convergence rate, and convergence point when considering the entire evaluation data. However, when we split the data into the high and low performing model groups, interestingly, the accuracy rate for high performing models is no longer in strong correlation with the convergence point. This can be partially explained by the fact that high performing predictive models are capable of making correct predictions even for action sequence examples that do not strongly follow the trained model (e.g., noisy data that are usually predicted as incorrect with low performing models), but they tend to begin to make correct predictions in a relatively later phase of the action sequence.

Conclusions

Automated player goal recognition is a key capability for enabling open-ended digital games to dynamically adapt gameplay experiences and promote players' engagement. In game-based learning environments such as CRYSTAL ISLAND, goal recognition can play a pivotal role in supporting tailored pedagogical scaffolding based on assessments of student learning and problem solving, and also inform game and curriculum design. This paper has introduced a data-driven goal recognition framework based on stacked denoising autoencoders that significantly outperforms the previous state-of-the-art technique. Empirical evaluations suggest that deep learning holds great potential as a novel computational approach to goal recognition for open-ended games. With accurate goal recognition models in hand, a promising direction for future work will be to design player-adaptive games that leverage the goal recognition models to create highly effective gameplay that is customized to individual players.

Acknowledgements

This research was supported by the National Science Foundation under Grant IIS-1138497 and Grant IIS-1344803. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. 2007. Greedy layer-wise training of deep networks. In Scholkopf, B., Platt, J., and Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems*, 19: 153-160. MIT Press.

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.

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.

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.

Deng, L., and Yu, D. Eds. 2014. Deep Learning: Methods and Applications. Now Publishers Inc.

Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1-30.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P. A., Vincent, P., and Bengio, S. 2010. Why does unsupervised pre-training help deep learning?. *The Journal of Machine Learning Research*, 11:625-660.

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

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.

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 International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 32-39.

Hinton, G. E., Osindero, S., and Teh, Y. W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7): 1527-1554.

Hinton, G., Salakhutdinov, R. 2006. Reducing the dimensionality of data with neural networks. Science, 313(5786): 504–507.

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.

Jennings-Teats, M., Smith, G. and Wardrip-Fruin, N. (2010). Polymorph: A Model for Dynamic Level Generation. In *Proceedings of the 6th Artificial Intelligence and Interactive Digital Entertainment Conference*, 138–143. Menlo Park, CA: AAAI Press.

Johnson, W.L. 2010. Serious use of a serious game for language learning. *International Journal of Artificial Intelligence in Education*. 20(2): 175-195.

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.

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.

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.

Li, B., and Riedl, M. O. 2010. An Offline Planning Approach to Game Plotline Adaptation. In *Proceedings of the Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, 45-50. Menlo Park, CA: AAAI Press.

Mateas, M. and Stern, A. 2005. Structuring Content in the Façade Interactive Drama Architecture. In *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, 93-98. Menlo Park, CA: AAAI Press.

Min, W., Rowe, J., Mott, B., Lester, J. 2013. Personalizing Embedded Assessment Sequences in Narrative-Centered Learning Environments: A Collaborative Filtering Approach. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence in Education*, 369-378.

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.

Palm, R. B. 2012. Prediction as a candidate for learning deep hierarchical models of data. Technical University of Denmark, Palm.

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.

Roberts, D. L., Cantino, A. S., and Isbell, C. L. 2007. Player Autonomy Versus Designer Intent: A Case Study of Interactive Tour Guides. In *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference*, 95-97. 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.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D. 1995. Artificial intelligence: a modern approach (Vol. 2). Englewood Cliffs: Prentice hall.

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

Shaker, N., Yannakakis, G. and Togelius, J. 2010. Towards Automatic Personalized Content Generation for Platform Games. In *Proceedings of the Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, 63-68. 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, 167-198. Cambridge, MA: The MIT Press.

Svozil, D., Kvasnicka, V., and Pospichal, J. 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1): 43-62.

Thue, D., Bulitko, V., Spetch, M., and Webb M. 2010. Socially Consistent Characters in Player-Specific Stories. In *Proceedings of the Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, 198-203. Menlo Park, CA: AAAI Press.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096-1103. ACM.

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