Revisiting RumbleBlocks with Apprentice Learning:

Examining and Learning Gameplay with AL

Cayden Codel **(ccodel@andrew.cmu.edu)**

Advised by Dr. Erik Harpstead and Dr. Ken Koedinger, Human-Computer Interaction Institute

WHAT IS RUMBLEBLOCKS?

RumbleBlocks is an educational game designed to teach young children how to construct stable towers. Once a tower has been constructed, an earthquake shakes the ground, testing the tower's stability.

Figure 1: A screenshot of gameplay[1]. The tower must be tall enough to cover the blue energy "checkpoints" and reach the alien and stable enough to support the UFO through an earthquake.

WHAT IS APPRENTICE LEARNING?

- Apprentice learning is a type of machine learning where agents receive correctness feedback from a teacher.
- Weitekamp, et al.[2] and MacLellan[4] developed a computational model for student learning called AL that can succeed at the same rates students do at mathematical tasks, such as fraction addition, in an intelligent tutoring system environment.
- AL is powered by TRESTLE[3], an algorithm that constructs a hierarchical categorization tree for concept formation.

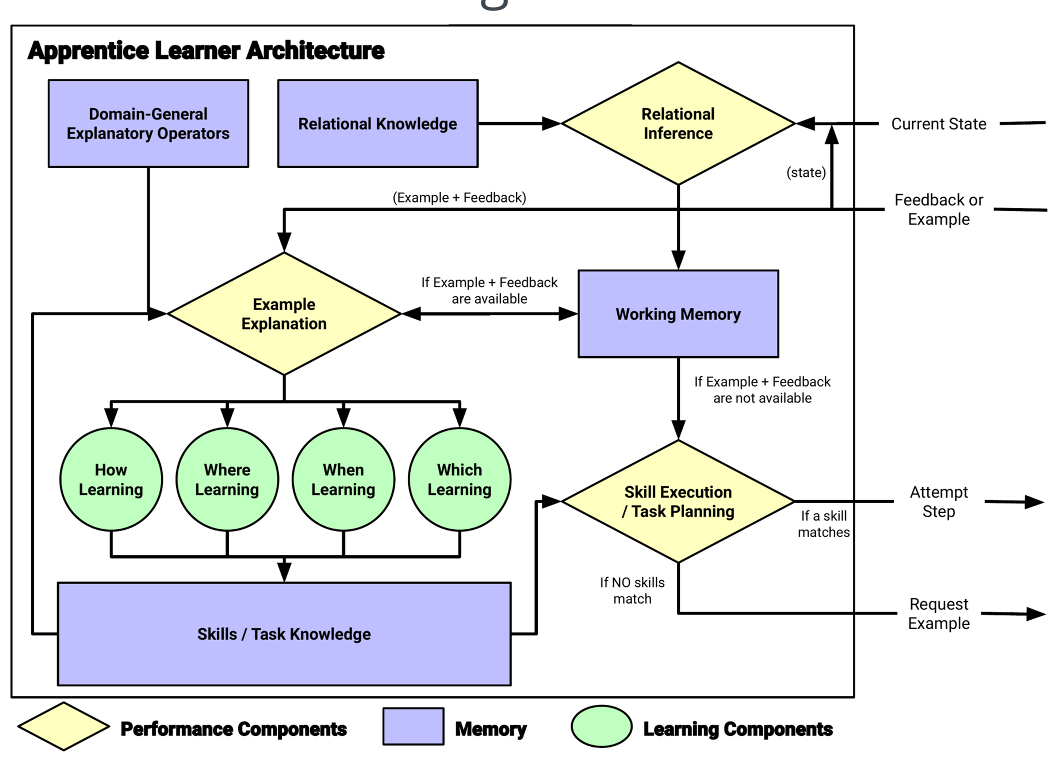

Figure 2: AL's architecture[4]. Through demonstrations and feedback, AL uses how-, where-, when-, and which- learning to generate the skills needed to solve problems.

AL IN RUMBLEBLOCKS

- Previous research shows AL learning to categorize stable buildings, but not how to construct them[3]
- Replaying human gameplay serves as a way to teach AL[5][6]
- Goal is to get AL to act like humans if given unseen levels
- At each time step, game state is serialized and sent to AL
- AL processes game state and attempts to make an action
- AL operates in a defined action space
  - Agents can move and rotate blocks and the UFO
- AL given prior knowledge in a set of skills/operators
  - E.g. `place_block_on_checkpoint` will place a building block directly on an energy checkpoint
- More advanced rules can allow for faster learning
  - E.g. `place_block_above` lets AL use orthogonal spatial relationships without learning them first

METHODS OF TEACHING AL

- Train on previously recorded human gameplay
  - Sequence of human actions on a level serve as the "teacher," with that human's success or failure on the level either "correct" or "incorrect"
  - If AL wishes to act, it may do so and receive reward
  - If AL takes no action, the next recorded action plays
  - At the end of the level, AL trains on all actions taken
- Interactive training
  - If AL wishes to act, it does so and asks a human if the action taken is correct
  - If AL takes no action, it asks the human to demonstrate
    - The demonstration then becomes the "correct" action
- Supervised learning
  - AL acts until level completion or an action limit is hit
  - Each action gives AL reward
  - Good for reinforcement learning agents, but not AL

INITIAL RESULTS

- Sending raw game state confuses AL
  - Untreated game state has floating point precision issues
  - Solution: round off position and rotation vectors
  - Led to faster training and higher usage of prior skills
- Many gameplay sequences do not demonstrate "mastery"
  - Mismatch between "good" and "ideal" towers
  - Additional or spurious actions serve to slow down AL
  - Interactive training per level may solve this, but time-consuming and requires human intervention
- Reducing number of prior skills lets AL solve some levels faster, leaves AL unable to solve others
  - E.g. `place_block_above` alone does not allow for placing a block offset from center
- Reward shaping in some cases helped
  - Can be used to incentivize placing blocks on checkpoints
    - Not all levels have checkpoints
  - If a loop is detected, repeated actions are punished

Future work may focus on vetting/treating human gameplay before giving it to AL, automating interactive training, modifying AL to handle floating-point values, and comparing AL's performance to other agents (e.g. RL).

WORKS CITED

[1] Harpstead, Myers, Aleven. In Search of Learning: Facilitating Data Anaysis in Educational Games. CHI Journal, 2013. https://dl.acm.org/doi/pdf/10.1145/2470654.2470667/

[2] Weitekamp, et al. Toward Near Zero-Parameter Prediction Usign a Computational Model of Student Learning. Journal of EDM, 2019.

[3] MacLellan, Harpstead, Aleven, Koedinger. TRESTLE: Incremental Learning in Structured Domains using Parial Mapping and Categorization. Advances in Cognitive Systems, 2015.

[4] MacLellan, Christopher. Computational Models of Human Learning. Unpublished thesis, 2017. https://chrismaclellan.com/media/publications/MacLellan-Camera-Ready.pdf/

[5] Holdmgard, Green, Liapis, Togelius. Automated Playtesting with Procedural Personas through MCTS with Evolved Heuristics. IEEE, 2019. Vol. 11, no. 4.

[6] Tychsen, Canossa. Defining Personas in Games Using Metrics. Future Play, 2008.