## CS-4701 / 5701 Project Ideas

- Explore reasoning capabilities of LLMs (eg. GPT-5). Can we quantify them?

- Consider basic the framework for AlphaGo (for Go) and AlphaZero (for chess) using OpenSpiel and experiment with learning, search, and play strength tradeoffs.

  Basic question: Does AlphaZero or AlphaGo have hidden weaknesses a human could exploit?

- *Ultimate* Tic-Tac-Toe (or "super Tic-Tac-Toe"), 3D Tic-Tac-Toe, or HEX (board game).

- Image recognition using Deep Neural Nets. Generative Adversarial Networks (GANS). (But go beyond what is already studied widely.)

- Explore the knowledge captured in GPT-4. How much common sense? What is missing?

- Implementing Virtual Predators-Prey in a Virtual Environment (Genetic Algorithms)

- Learning to Play Checkers

- Apply a genetic algorithm to the problem of automatic generation of computer programs. Contrast with GPT-4 capabilities.

- Build a system that uses heuristic search (with minimax and alpha-beta pruning) to play Connect-4. Evaluate it through self-play against versions with different heuristics and search effort. Add learning to tune heuristics. Determine how close to optimal the learned system plays. Optimal strategy is known for Connect-4.

- Build (and train) a system that plays Connect-4 using a deep neural network. Train using minimax player. Evaluate performance.

- A chess endgame player. An interesting variant is to design a method that learns end-game rules from examples and compare it with hand-generated chess endgame players.

- A theorem-proving system for some (small) subset of mathematics.

- A program that generates automatic crossword puzzles, starting from a dictionary and an empty board.

- Recreate from its specifications the reinforcement learning (neural net) system that learns to play backgammon by playing games against itself (Tesauro, 1992).

- A reactive, rule-based system that plays Tetris.

- I also have several more research-oriented projects. If interested, email me at selman@cs.cornell.edu. Use CS-4701 in Subject. For these projects, I will provide more direct guidance on research questions that can feasibly be explored for a course project. We will build on previous project work. So, you don't need to start from scratch. Examples next page.

**Example topics:**

(1) Explore what common sense knowledge is captured and what is still missing in GPT-4 or GPT-5.

(2) What are the reasoning capabilities of GPT-4 and GPT-5? How much do train of thought or tree of thought approaches add?

(3) How does GPT-4 or GPT-5's understanding of language differ from human understanding?

(4) In what sense is GPT-4 or GPT-5's intelligence different from human intelligence?

(5) AlphaGo and AlphaZero are amazing at selecting the best or near-best possible moves from a set of legal moves. However, what about learning what a legal more is? Can deep Reinforcement Learning be used to feasibly learn the rules of a game? For us, learning to play chess is definitely easier than becoming a world champion chess player. But is this also true for a deep learning system? The question is related to DeepMind's work on MuZero. Core question: Did MuZero really learn the legal moves for chess from just observing games?

(6) Mathematical discovery. Can an AI system discover some interesting mathematical facts from the basic axioms of number theory? Can AI display mathematical creativity? (Guidance from Bart Selman.)

(7) Explore the proof of Erdos Discrepancy Conjecture. What information is discovered during the reasoning process? (Guidance from Bart Selman.)