Christopher Lee Simpkins

Research Statement

1 Current Focus and Future Directions

I love to build software, I have a passion for natural and artificial languages, and I am deeply interested in intelligent systems. These three perspectives converge in my current focus: creating the building blocks to enable working programmers to build AI systems. In my doctoral research I integrated reinforcement learning (RL) into an experimental programming language and ran a programmer study to validate the usefulness of doing so. This language, AFABL (A Friendly Adaptive Behavior Language), is a domain-specific language embedded in Scala and includes abstractions for direct expression of RL problems. To make AFABL work as envisioned I needed to invent a new modular RL algorithm that did not suffer the reward scale coupling problem inherent in the existing state of the art. I presented this new formulation of modular RL and an algorithm that implements it at AAAI 2019.

There are two growing trends that fall directly within my primary research interests. First, there is growing consensus that reinforcement learning will be the "next big thing" in AI/machine learning. The second trend is bridging the gap between AI/machine learning research and production systems (where "production" may be broadly construed). I envision a future in which practitioners – working programmers and researchers in specialties other than AI – can harness the power of RL that is well integrated into software development systems and processes. Realizing this vision will require coordinated advancements in

- **modular reinforcement learning.** My recent work in composable modular RL solved the problem of reward scale coupling in modular RL agents using tabular RL algorithms. Future work will investigate different RL algorithms, including deep RL, action selection negotiation, goals/reward specification, and unification of modular and hierarchical RL.
- **AI programming systems.** One of my favorite features of programming language research is that it puts CS research in the hands of working programmers. For example, few working programmers have any knowledge of formal methods, but every time a Java programmer writes a static type annotation that programmer is employing formal methods (type safety verification). Injecting research results into programming languages is a particularly fruitful way to bring the benefits of CS research to software practitioners. As I expounded for OOPSLA Onward! 2008, my AFABL DSL integrates RL into an AI programming system in the same way that Prolog integrates backtracking search in a logic programming system. Future work will include integrating additional AI algorithms, better integration with compiler and interpreter tool chains, and designing programming abstractions for **AI system architecture.** Modern AI systems tend to integrate many approaches. For example, DeepMind's Alpha Go combined deep neural networks, supervised learning, Monte Carlo tree search, and RL. AI programming systems need to support AI system architecture in the same way that object-oriented programming supports business system architecture or functional programming supports distributed concurrent system architecture.
- **scaling AI programming.** Modern machine learning frameworks have a reputation for being difficult to use in practice, and one critique of Google's recent successes is that Google has

access to massive amounts of purpose-built computing infrastructure. But many organizations already employ qualified programmers without specialist machine learning knowledge and use big data platforms like Hadoop and Spark. We should think early in the research process about bringing scalable machine learning to the practical computing platforms and staff of existing organizations. Whenever practicable we should design AI algorithms with the execution models of big data systems in mind, for example, by employing abstractions from functional programming and data-parallel collections.

- **integrated intelligence.** I have always been interested in integrating knowledge-based approaches to AI with modern quantitative methods. Integrated intelligence is explicitly enabled by my modular RL work, and I have published work in integrating RL with cognitive modeling at the International Conference on Cognitive Modeling in 2010. The Computing Community Consortium and the NSF is creating a 20-year roadmap for AI research, including a massive commitment of funding on the order of billions of dollars. I attended a panel led by some of the creators of the roadmap at AAAI-2019. One of the primary research thrusts of the 20-Year Roadmap is integrated intelligence.

2 History

While I focus on specific areas, I enjoy solving problems wherever they arise. There is a great deal of opportunity in seeing connections between problems in one area and solution methods from other areas. I began my research career at Georgia Tech Research Institute (GTRI), the applied research arm of Georgia Tech in a lab that specialized in Physics and Electrical Engineering. During my time at GTRI I created interactive visualizations for physicists working on magnetic field simulations, used optimization algorithms to design neural networks for specific emitter identification, collaborated on a proposal for design software for quantum computers (think VHDL compiler that emits quantum instead of classical logic circuits), and applied the MIMIC (Mutual Information Maximizing Input Clustering) algorithm co-invented by my advisor, Charles Isbell, to the design of planar array antennas. A group in my lab had been using a Beowulf cluster to run antenna simulations that implemented the evaluation function for a genetic algorithm (GA). The GA often stalled in local maxima and needed to be "re-seeded" with new candidates hand-designed by electrical engineers, and since the evaluation function was expensive the GA needed weeks to produce designs. I thought that since MIMIC discovers and exploits patterns in the structure of candidates it might solve these two key problems. MIMIC is far less susceptible to local optima, and MIMIC requires orders of magnitude fewer iterations to converge to a solution. I led a team of three junior researchers who implemented a MIMIC-based algorithm and found it far superior to the GA on a test data set. I am preparing a paper reporting these results for NeurIPS 2019.

During my final years at GTRI I created a collaboration between our lab and the College of Computing and was heavily involved in funding development. I participated in proposers' day workshops at DARPA, worked with AFRL, and co-led a proposal for a multi-million dollar, multi-institute contract from IARPA for proactive intelligence. I assembled a team that included an internationally renowned terrorism expert, Marc Sageman, an intelligence analysis expert, Stephen Marrin, and forged working relationships with an intelligence analyst in the Defense Intelligence Agency and program managers at AFRL. We narrowly lost that proposal in the final round. Nevertheless, I learned a great deal about contract development and secured funding for several smaller projects.