Teaching Statement

Daphne Ippolito

The opportunity to mentor and educate students is the primary reason I am pursuing a career in academia rather than in industry. Computer science departments have a vital role in universities that goes far beyond teaching courses to CS majors. Programming and analytical reasoning skills are increasingly necessary tools for success in many fields, from the digital humanities to biology and the social sciences. My goals as an educator are first to communicate my own excitement about computer science and its many applications and then to equip students with the CS knowledge they need to be successful whatever their chosen endeavor.

Scaling Computer Science Education

The current reality is that growth in interest in CS courses is outpacing the capacity to teach every student in traditional small classroom settings. Because of high demand, we either have to restrict enrollment or accept class sizes too big for much individual attention. I am in favor of scaling our classes to meet the demand while adapting and innovating on traditional teaching methods to create a positive learning experience. Interactive and collaborative classrooms and the use of rapid feedback mechanisms will help me to accomplish this.

In 2020, I taught a graduate-level topics course on language generation where I put these concepts into practice. I made frequent use of a polling website called Mentimeter to allow students to answer questions on their phones with the results transmitted live to my screen. In addition to testing their comprehension of key material, I also polled students for what topics they were most interested in, solicited feedback on homeworks, and ran ice-breaker activities. When presenting particularly hard subjects, I set it up so students could type in questions. In a large classroom, students might be daunted to raise their hand and ask a question, or worse acknowledge their lack of comprehension, but I found that having an avenue to anonymously ask and upvote questions live during the lecture lowered the barrier to classroom participation. It also allowed me to batch questions and answer at logical breaking points, leading to a more coherent and interruption-free presentation of the material. In an end-of-class survey, all but one student answered “yes” that use of interactive polling helped keep them engaged, and the one “no” conceded that “It wasn't too fun, but in retrospect probably did force me to pay attention at certain points.”

The design of homework is another area that must necessarily change to scale to larger class sizes. Computer science as a discipline has the unique ability to provide near-immediate automated feedback to students, in that their code can be run and the results checked for correctness. However, while automatic graders may teach students to write *correct* code, they can only do so much for teaching students to write *good* code. In an ideal world, instructors would go through each students' submission and check for detailed comments, succinct implementations, and other elements of good code. When this is not possible, I would like to experiment with assignments asking students to review each other’s code. Peer reviews are not only a way to provide students qualitative feedback, but they also offer an opportunity to teach students *how* to give useful feedback. Effective written communication is a skill that is sometimes undervalued by CS majors but is invaluable for most software engineering jobs, where giving and responding to code and design reviews is standard procedure.

Teaching Experience

My thoughts on pedagogy have been heavily influenced by the diverse set of classes I've had the privilege to TA. At the undergraduate level, I've TA'd *Computational Thinking,* a class which introduces CS to non-majors, sophomore-level *Systems Programming*, and an elective *Introduction to Robotics*. At the graduate level, I've TA'd M*achine Perception* and *Computational Linguistics*. All but one of these classes had more than 50 enrolled students.

In Spring 2020, I developed and taught a Special Topics course on *Interactive Fiction and Text Generation* along with my advisor, Chris Callison-Burch. The class attracted about 25 students with very diverse interests and abilities, and my thoughts on an interactive classroom were heavily influenced by the challenge of keeping these students engaged. In the end-of-class survey, students said: "Daphne was amazing throughout! She is very approachable for help, and is very knowledgeable in the field." and "I found Daphne's teaching style to be really engaging! She was definitely able to stimulate interest in the topic." I especially appreciated this feedback given the challenge of transitioning the class from an active learning classroom to Zoom halfway through the semester.

At the University of Pennsylvania, I am currently completing the requirements for a Certificate of Teaching from the Center for Teaching and Learning, which has involved attending workshops on pedagogy and having a lecture observed and reviewed. The workshops have covered topics that include bringing lessons from online learning to physical classrooms, strategies for collaborative experiential learning, and inclusive teaching.

Teaching Interests

While I would be very happy to teach upper-year undergraduate or graduate-level courses in natural language processing, I am especially interested in teaching an introductory computer science course. In my opinion, introductory classes are some of the most challenging to teach. Concepts like recursion and variable scope are second-nature to us who've been programming for decades, which makes them all the more difficult to communicate to students. I aspire to have the skill of taking engrained understanding and transmitting it to newcomers for whom it is not yet intuitive. I would also like to take advantage of the fact that natural language processing, my area of expertise, more than many other areas of computer science is accessible to novice programmers. It is possible to perform meaningful analyses of textual data using fairly basic algorithms and data structures. I think that incorporating methods from NLP into the teaching of early-year undergraduates will help get students excited about applying their learning to real-world data and problems.

Finally, I am also interested in teaching smaller seminar classes at the graduate level. My topics class on text generation was popular enough at Penn that it is going to be taught a second time, and I imagine a similar course would be popular at University of Toronto as well. I would also be interested in teaching more narrowly focused Special Topics courses, for example: (1) a course on human and automatic evaluation methods for text generation systems, and (2) a course on language modelling, starting with statistical models and progressing to state-of-the-art neural language models. Both of these are courses I wish I could have taken!

Academic Mentorship

At Penn, I've had the privilege to be able to mentor both undergraduate and Masters students on independent research projects. Since 2018, I have mentored about two students per semester on undergraduate research projects or Masters' theses. I've especially appreciated the opportunity to work with some students over the course of their undergraduate careers, allowing me to track their growth from novice software engineers

needing clear instructions to competent, independent researchers with their own taste in problems. For example, in 2018, a second-year undergraduate assisted me in building ChatEval, a tool for evaluating dialog agents. The same student took my topics course in 2020, and their final project turned into RoFT, a web-based game for evaluating NLG systems by asking humans to detect machine-generated text. Both projects were published at top conferences.

As an assistant professor, I look forward to mentoring students at all stages, not just PhD students, and to building larger research collaborations with teams of students at differing skill levels and seniorities. I plan for my PhD students to learn not just how to conduct independent research but also how to mentor others in conducting research, a skill my Penn advisor has done an excellent job teaching me. Lastly, I strongly believe in research agendas being set bottom-up. While I am prepared for students who require assistance in identifying promising directions, I am most looking forward to guiding the students who come to me with projects or research questions of their own design. Fostering creativity and independent thinking is an essential role of a good teacher.