Diversity and Inclusiveness in Teaching and Research

I am passionate teaching and mentoring students from all backgrounds, and I approach this in a number of ways. One of my focuses is on expanding access to computational methods for students of all backgrounds. Another belief of mine is that creating an inclusive learning environment requires that instructors know their students well. I also believe that fairer pedagogy involves being explicit about expectations. My teaching and mentoring philosophy encompasses four core principles:

- **Interactive, participatory learning**: understanding student needs through interactive pedagogy
- **Multidisciplinary learning**: integrating insights from other fields enriches student learning.
- **Fairer pedagogy through uncovering the hidden curriculum**: expectations and foundational assumptions should be made explicit. This helps students from diverse backgrounds to succeed.
- **Mentoring students from diverse backgrounds**: structural efforts can only go so far. We also must reach out to promising students who might otherwise lack access to computational methods.

Interactive learning

As an instructor and as a mentor, I prioritize getting to know my students: their goals, their aspirations, and their prior experiences. Students come into college with different experiences and different backgrounds. These past experiences shape their needs as learners, and understanding the full context of students' experiences is a precursor to designing curriculum that meets their needs. I find that being able to contextualize each student helps me adapt the course content to their needs and interests.

One way to get to know students is to teach highly interactive classes. I design my classes to be highly interactive, so that I have a chance to get to know my students throughout the class. This takes different forms depending on the class: when I have taught computational linguistics, I incorporated frequent brief in-class exercises, while in linguistics theory classes, group work or discussions work well. This allows me to observe how well students are learning and tailor my plans.

Multidisciplinary learning

Statistical and computational skills can open up new possibilities for linguistics research. At the same time, the social sciences have a lot to offer computer science students: they provide the rigorous training in experimental design and data validity and insight into the questions of ethics and fairness. I believe that strengthening the connections between disciplines not only improves the quality of education for students, but can lead to a fairer world by giving future technologists grounding in ethics and equipping a more diverse set of students with the expertise they need to participate in the development of new technologies.

There is a growing emphasis on experimental, computational, and fieldwork methodology in linguistics, and I believe that it is important to equip students with the methodological tools that they need to pursue innovative, exciting research. I am particularly passionate about making computational and statistical methods accessible to students who may not have had much previous exposure to quantitative methods. I think this is important both for improving the diversity of computational subfields and for enabling students from all backgrounds to pursue the research they care about.

While at UMass, I have worked to strengthen the already strong ties between the Linguistics department and the CS department. I have taken classes in the CS department alongside CS graduate students, and have helped CS graduate students in Linguistics classes, as the TA for Cognitive Modeling. I have attended reading groups in the CS department, in particular, the NLP reading group. I have also started a new reading group in the Linguistics department which focuses on neural networks for linguistics.

Although designed to run for a single semester, the Neural Networks in Linguistics reading group has been meeting weekly for over a year. Graduate students from both the Linguistics and CS departments have presented their research, and we have read a variety of papers focused on using linguistic expertise to probe neural networks, using neural networks to contribute to theoretical linguistics, and designing linguistically-

informed neural networks to for NLP. The attendance is evenly split between faculty and students from the Linguistics department and the CS department. In order to make the group approachable for students with little computational background, we ran a tutorial on fundamental concepts in deep learning.

Fairer pedagogy

I believe that it is important to make the "hidden curriculum" visible: the important academic assumptions and practices that are rarely made explicit. I am currently teaching a first year seminar on Technology for Language Revitalization. Along with the core content of the course, I am also helping students transition into the college community. For instance, we discussed writing professional emails. This was helpful on two different levels. First, many students had been taught unhelpful rules in high school, such as that contractions shouldn't be used in formal writing, which led to a discussion of language ideologies that fed into the core content of the class. Second, many students were unsure of how to contact professors respectively, and expressed a lot of anxiety about getting it wrong, even though this is a skill that many professors take for granted. Making these expectations explicit can help students from all backgrounds succeed.

Advocating for students

I have also worked within the Linguistics department to mentor students from backgrounds that are underrepresented in computer science that are interested in pursuing computing careers. Through working with my undergraduate research assistant, I have come to see how there are both personal and structural solutions to the diversity problem.

My research assistant, Tessa, had no prior experience in computing when they came to college, but is now an excellent programmer. Nonetheless, when they interviewed for a computer science internship this past summer, the team was very reluctant to hire them because they didn't have experience with Unix. Despite having taken half a dozen CS courses at two universities, they had never been taught how to use the command-line, because this was seen as too basic to cover in a computer science class. For lack of this basic bit of knowledge, which could have been covered in a single lecture, Tessa was almost passed over for an opportunity that they ended up excelling at.

There are two lessons that I draw from this. First, it is important to teach the hidden aspects of computing explicitly: if we want to promote a diverse computing community, we cannot make assumptions about our students' backgrounds. Whenever possible, we should work to minimize the gaps between what students are taught and what they are expected to know. This is not to say that classes cannot be rigorous; they can and should be. I try to design classes to start from the very basics, but to increase quickly in difficulty. This is the approach that I took when teaching non-CS majors as the TA for Cognitive Modeling, and I saw that it allowed good students from all backgrounds to excel.

Second, as individuals, we should look for these gaps and work to remedy them. As a graduate student, I wasn't able to re-structure the curriculum to make it more inclusive of students from all backgrounds. But I was able to advocate for Tessa and help them learn command-line basics before the start of their internship.

Progress in diversity comes about through a combination of personal and structural changes. As computational community members, we can work to build ties with other departments and to welcome their students into the computational methods community. As instructors, we can advocate for teaching practices that are inclusive. As mentors, supervisors, and peers, we can work with individuals who have not been well-served by the computational community in order to broaden access to computational methods.

1 Summary

I am strongly committed to nurturing a diverse, inclusive campus community. I aim to broaden access to computation methods in the following ways: (1) unveiling the 'hidden curriculum' that disadvantages students from non-traditional backgrounds; (2) building interdisciplinary connections to broaden access to computation methods; and (3) advocating and mentoring students from diverse backgrounds.