Teaching Statement

Ishani Mondal

Scientific knowledge is only useful once it reaches someone who can act on it, and teachers carry it across that gap. I want to train students who can build computational systems and also make those systems legible to the people they serve—clinicians, scientists, and readers who will never see the model but will live with its output. My research asks how an explanation reaches a particular person; teaching is the same question asked in a room rather than a system.

I learned early that access to education is not evenly distributed. My father's own schooling was hard-won, and what it cost him is the reason I have never been able to treat a classroom as a neutral space that everyone enters on equal terms. Some students arrive having been taught how school works. Others are learning that at the same time as the material.

Three teachers shaped how I think about the work. Prof. Sudeshna Sarkar at IIT Kharagpur, for whom I was a teaching assistant across four course offerings, taught the same core ideas to first-year programmers, to graduate researchers, and to a national online audience, and watching her do it showed me that the material is not the course—the audience is. My PhD advisor, Prof. Jordan Boyd-Graber, teaches in a way I had never encountered: his QANTA project runs on incremental question answering, where a question is read aloud and players interrupt the moment they believe they know the answer, so what gets measured is not whether you can produce an answer at the end but how early, and on what evidence, you are willing to commit. From him I learned that a classroom is more honest when students are asked to commit before they are comfortable. And Dr. Kalika Bali at Microsoft Research India showed me that care and rigor are not competing demands on a mentor. She was generous with attention and immovable about depth, and she made clear that taking a student seriously means holding their work to a real standard rather than protecting them from one.

My teaching philosophy is to ground methods in problems whose stakes students can see, to build classrooms where participation does not depend on prior fluency, and to mentor students toward the independence to ask their own questions. These commitments also come from the research that formed me: building a linguistic annotation framework with multilingual HIV-positive youth in a peer support forum in Kenya, and working on speech technology for Mundari, a language with almost no computational resources. In both cases something true and useful existed, and the people who needed it most could not reach it. Every student who enters a classroom is a version of that problem, and each arrives with a different distance to travel.

Grounding Methods in Problems Whose Stakes Students Can See

Students engage with a method when they can see what breaks if it fails. In 2025 I taught natural language processing at the University of Maryland with Prof. Rachel Rudinger, where I developed course materials and mentored student projects. Writing materials taught me that my instinct for explanation runs in a predictable direction: my first drafts ordered ideas by the logic of the finished result, with the intermediate reasoning collapsed—the order that makes sense to someone who already understands. I rewrote by dependency instead, asking of each unit what must be true in a student's head before this sentence can land. The clearest case was the unit on evaluation: rather than introducing BLEU and ROUGE as formulas and then discussing their weaknesses, I had students first hand-score a set of model outputs themselves and argue about the disagreements, so that when the metrics appeared they were recognizable as attempts to automate a judgment the students had already made and already found hard.

I bring the same commitment to teaching evaluation, the part of the field I know best and the part students are least often taught. My work asks whether a measurement detects understanding or only fluency: PEDANTS scores answer equivalence in a way that tracks human judgment better than exact

match or BERTScore at lower cost than an LLM judge, and AdvScore measures whether a benchmark is hard for models yet easy for people. In a classroom this becomes a habit of asking students not only whether their system works but how they would know, and what evidence would change their mind.

Beyond formal courses, I have organized two weekly reading groups at Microsoft Research India since 2021, in speech and language processing and in user-centered technology for emerging markets, including inviting external speakers. Choosing what a mixed-seniority room can read productively, week after week, has taught me as much about pacing and prerequisites as any course.

Building Classrooms Where Participation Does Not Depend on Prior Fluency

Much of what makes a course hard for some students is not the material but the conventions surrounding it, and only the material is on the syllabus. In *Introduction to Programming and Data Structures* (CS10001, with Prof. Aritra Hazra) at IIT Kharagpur, the students were first-years, and the obstacle was rarely the concept: it was what an error message is for, how to isolate a failing case, the difference between a program that runs and one you understand. In *Introduction to Deep Learning* (CS60010, with Prof. Sudeshna Sarkar) the students were graduate researchers, and the useful move was the opposite—withdraw scaffolding and ask whether they could defend a design rather than reproduce one. Teaching both taught me that pitching to the median serves no one, and that naming a convention aloud costs a few minutes and removes a real barrier.

I try to structure participation so that it does not simply reward confidence. Open questions to a room reliably surface the same few students, which gives an instructor a comfortable and misleading picture of what the class understands. I use brief written responses and anonymous polls so that every student commits to an answer before anyone speaks, which gives quieter students a position to defend rather than an improvisation to perform. This practice is supported by evidence that active learning improves performance in STEM courses broadly and narrows achievement gaps for underrepresented students in particular. I also take seriously the finding that students in active classrooms often *feel* they have learned less even as they learn more which is why I explain to a class why I teach the way I do rather than letting the discomfort go unaddressed.

This is the practice I took from Boyd-Graber's incremental format: the useful signal is what a student will commit to before they are certain, and what they do when the commitment turns out wrong. His group is also a model of who participates, bringing undergraduates alongside graduate students from computer science, language science, and information science onto the same systems.

Mentoring Students Toward Independence

Mentorship is the part of teaching I care about most, and Kalika Bali's example—attentive and exacting at once—is the one I try to follow. Two recent students illustrate how differently that has to be applied. Javad Baghirov worked with me on FigTalk, our system for generating paper-grounded video explanations of scientific figures, and took ownership of the figure-decomposition stage that produces the validated inventory of regions the narration is permitted to point at; he began with a bounded, well-specified component because early traction is what carries someone through the stretch of a first project where nothing works, and finished it working largely independently. Meera Bharadwaj came to SMART-Editor with more preparation, so we started open-ended and negotiated the framing together, and she took on the invariance checks that verify an edit changed what was intended and nothing else.

In both cases I began from what the student wanted rather than from what I needed done, and in both I handed over a genuine open question rather than an exercise with a known answer, because the experience of not knowing whether something will work is the thing being taught and cannot be simulated. The judgment this requires is calibration: a week spent on an approach that fails is sometimes the most valuable week of a project and sometimes only discouraging, and which one it is depends on the student and the stage.

Because my work spans NLP, HCI, and scientific domains, much of what I teach students is how to move between fields whose conventions are never stated aloud—what counts as evidence, and which claims need a citation rather than an experiment. I have also learned a great deal from my mentees, whose questions have repeatedly sent projects in directions I would not have chosen. Meera's observation that our evaluation only ever asked what an edit had changed and never what it had preserved is what moved SMART-Editor toward explicit invariance constraints, which became the paper's central contribution.

Teaching Plan

I am comfortable teaching undergraduate and graduate courses in natural language processing (CMSC 470 and CMSC 723 or their equivalents), machine learning, deep learning, and information retrieval, and I have taught students ranging from first-year programmers to graduate researchers across computer science and language science.

I would be excited to develop two courses. The first is on the evaluation of generative systems: students would begin by breaking a published metric, constructing inputs on which it certifies something plainly wrong, then work through measurement design, annotation and agreement, human-grounded difficulty, and the failure modes of LLM-as-judge, before building and validating a measure of their own. Evaluation is under-taught relative to modeling, and students who cannot tell whether a system helped anyone will keep building ones that do not.

The second is a project-based course on AI for scientific communication, in which students take a real paper from a partner domain—biomedical or public-health research, where my own work on clinical trials and healthcare communication is rooted—and produce a faithful explanation of it as a diagram, a slide deck, or a narrated walkthrough, then design an evaluation and run it on readers from the intended audience. The deliverable is not the artifact but the evidence about whether it worked. My research on figure-grounded video, structure-preserving diagram generation, and persona-aware slides supplies the technical scaffolding, and the course would teach the rarer lesson: discovering that something you believed was clear taught no one anything, and diagnosing why.