Feedback is an essential component of the learning process, but in fields like computer science, which have rapidly increasing class sizes, it can be difficult to provide feedback to students at scale. Intelligent tutoring systems can provide personalized feedback to students automatically, but they can take large amounts of time and expert knowledge to build, especially when determining how to give students hints. Data-driven approaches can be used to provide personalized next-step hints automatically and at scale, by mining previous students’ solutions.

I have created ITAP, the Intelligent Teaching Assistant for Programming, which automatically generates next-step hints for students in basic Python programming assignments. ITAP is composed of three stages: canonicalization, where a student's code is transformed to an abstracted representation; path construction, where the closest correct state is identified and a series of edits to that goal state are generated; and reification, where the edits are transformed back into the student's original context. With these techniques, ITAP can generate next-step hints for 100% of student submissions, and can even chain these hints together to generate a worked example. Early analysis showed that hints could be used in practice problems in a real classroom environment, but also demonstrated that students’ relationships with hints and help-seeking were complex and required deeper investigation.

In my thesis work, I surveyed and interviewed students about their experience with help-seeking and using feedback, and found that students wanted more detail in hints than was initially provided. To determine how hints should be structured, I ran a usability study with programmers at varying levels of knowledge, where I found that more novice students needed much higher levels of content and detail in hints than was traditionally given. I also found that examples were commonly used in the learning process, and could serve an integral role in the feedback provision process. I then ran a randomized control trial experiment to determine the effect of next-step hints on learning and time-on-task in a practice session, and found that having hints available resulted in students spending 13.7% less time during practice while achieving the same learning results as the control group. Finally, I used the data collected during these experiments to measure ITAP’s performance over time, and found that generated hints became more optimal as data was added to the system.

Thesis document can be found at kriwers.net/thesis.pdf