New model shows how often to review material for flashcard programs

January 26 2012, By Bill Steele

(PhysOrg.com) -- A challenge for students and teachers -- and today, for designers of educational software: How often should material be reviewed for best learning? Wait too long to review and it fades away; review too soon and the effort is wasted.

Tim Novikoff, Cornell graduate student in the field of applied mathematics, faced that problem when he created Flash of Genius, a smartphone app that displays vocabulary flashcards for SAT preparation. So he developed a mathematical model for educational software. The results are described in the online early edition of the Proceedings of the National Academy of Sciences the week of Jan. 23. Co-authors are Jon Kleinberg, the Tisch University Professor of Computer Science, and Steven Strogatz, the Jacob Gould Schurman Professor of Applied Mathematics.

The paper describes "how to schedule the introduction of new material and the review of material given a set of parameters that describe the student," Novikoff said. Most educational software, he explained, uses an arbitrary review schedule and just hopes it will be convenient for the user.

"The model is based on what the psychologists have been finding out about the process of learning, and we're hoping it can provide a language for new kinds of educational software," Kleinberg added.

Psychologists report that each time an item is reviewed, the learner can

go longer before needing to review it again. So a new item might be repeated four steps later, then eight steps after that, and so on. The "spacing constraints" will be different for each student. Ideally, software should observe how well each student retains lessons and develop rules to fit. But as the number of cards increases, this becomes harder to schedule -- even for a computer.

As more and more cards are in line for review, "There will be time steps where more than one card has to show up," Novikoff pointed out. "This model gets you around that but also shows that sometimes it's not possible."

The ideal goal defined in the paper is "infinite perfect learning," where new items can be added forever and every item is continually reviewed. An alternative is "cramming," where the student seeks to learn a finite list of items in a specified time period.

The paper suggests three ways of scheduling material for infinite perfect learning: the Recap Method, for fairly fast learning; the Slow Flashcard Method, which is what it sounds like; and Hold-Build, for learners who benefit from quick repetition of new material. But a few "finicky slow students" will require so much review that it becomes impossible to introduce new material. But it is possible to construct a cramming schedule for any student who can be described by the model.

The model is not meant as a cookbook for software developers, Novikoff noted, but rather as a framework that defines the spacing constraints of a theoretical student. Then it's up to the software developer to find the right constraints and plug them into the model. To develop algorithms (the underlying procedures for software, expressed in mathematical terms), a programmer must have a formal mathematical model to start with, Novikoff explained.

Very soon, he suggested, it may be possible to analyze data from students to develop an "average" set of constraints that educational software can use as a starting point from which it adjusts to fit each student.

The work is supported by the National Science Foundation, the MacArthur Foundation, Google and Yahoo.

Provided by Cornell University