Computer system automatically solves word problems

May 2 2014, by Larry Hardesty

This image shows a word problem provided by the researchers. The answer appears in the second image. Credit: Jose-Luis Olivares/MIT

Researchers in MIT's Computer Science and Artificial Intelligence Laboratory, working with colleagues at the University of Washington, have developed a new computer system that can automatically solve the type of word problems common in introductory algebra classes.

In the near term, the work could lead to educational tools that identify errors in students' reasoning or evaluate the difficulty of word problems. But it may also point toward systems that can solve more complicated problems in geometry, physics, and finance—problems whose solutions don't appear in the back of the teacher's edition of a textbook.

According to Nate Kushman, an MIT graduate student in electrical engineering and computer science and lead author on the new paper, the new work is in the field of "semantic parsing," or translating natural language into a formal language such as arithmetic or formal logic. Most previous work on semantic parsing—including his own—has focused on individual sentences, Kushman says. "In these algebra problems, you have to build these things up from many different sentences," he says. "The fact that you're looking across multiple sentences to generate this semantic representation is really something new."

Kushman is joined on the paper by Regina Barzilay, a professor of computer science and engineering and one of his two thesis advisors, and by the University of Washington's Yoav Artzi and Luke Zettlemoyer. The researchers will present their work at the annual meeting of the Association for Computational Linguistics in June.

Finding your place

The researchers' system exploits two existing computational tools. One is the computer algebra system Macsyma, whose initial development at MIT in the 1960s was a milestone in artificial-intelligence research. For Kushman and his colleagues' purposes, Macsyma provided a way to distill algebraic equations with the same general structure into a common template.

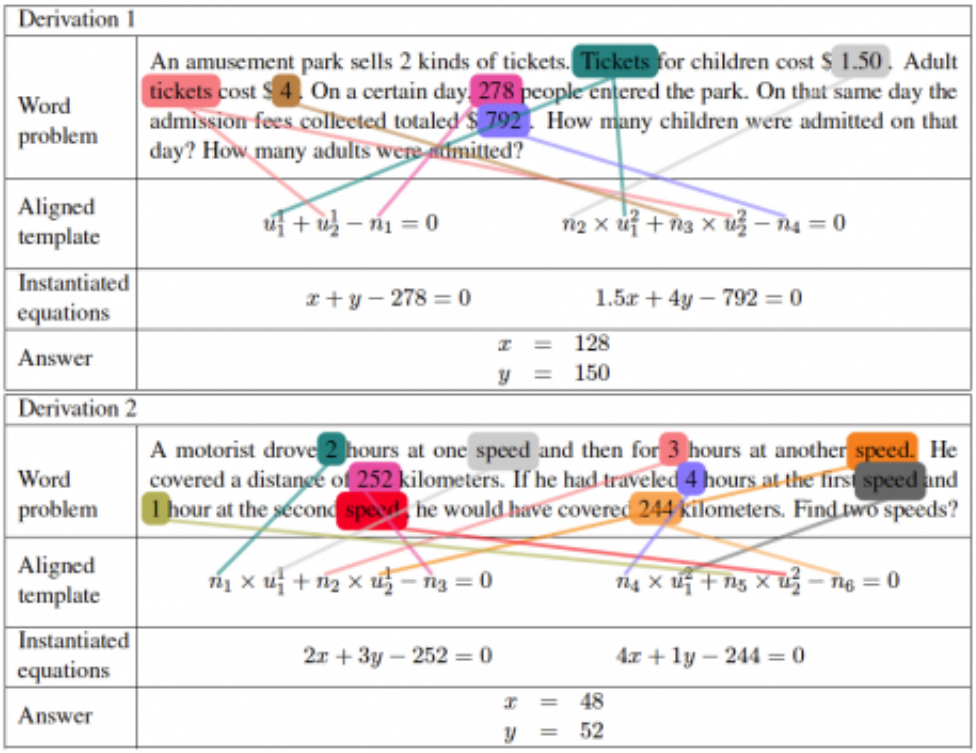

Figure 2: Two complete derivations for two different word problems. Derivation 1 shows an alignment where two instances of the same slot are aligned to the same word (e.g., u_1^1 and u_1^2 both are aligned to "Tickets"). Derivation 2 includes an alignment where four identical nouns are each aligned to different slot instances in the template (e.g., the first "speed" in the problem is aligned to u_1^1).

Credit: Courtesy of the researchers

The other tool is the type of sentence parser used in most natural-language-processing research. A parser represents the parts of speech in a given sentence and their syntactic relationships as a tree—a type of graph that, like a family-tree diagram, fans out at successive layers of depth.

For the researchers' system, understanding a word problem is a matter of correctly mapping elements in the parsing diagram of its constituent sentences onto one of Macsyma's equation templates. To teach the system how to perform that mapping, and to produce the equation templates, the researchers used machine learning.

Kushman found a website on which algebra students posted word problems they were having difficulty with, and where their peers could then offer solutions. From an initial group of roughly 2,000 problems, he culled 500 that represented the full range of problem types found in the larger set.

In a series of experiments, the researchers would randomly select 400 of the 500 problems, use those to train their system, and then test it on the remaining 100.

For the training, however, they used two different approaches—or, in the parlance of machine learning, two different types of supervision. In the first approach, they fed the system both word problems and their translations into algebraic equations—400 examples of each. But in the second, they fed the system only a few examples of the five most common types of word problems and their algebraic translations. The rest of the examples included only the word problems and their numerical solutions.

In the first case, the system, after training, was able to solve roughly 70 percent of its test problems; in the second, that figure dropped to 46 percent. But according to Kushman, that's still good enough to offer hope that the researchers' approach could generalize to more complex problems.

Featured performance

In determining how to map natural language onto equation templates, the system examined hundreds of thousands of "features" of the training examples. Some of those features related specific words to problem types: For instance, the appearance of the phrase "react with" was a good indication that the problem dealt with chemistry. Other features looked at the location of specific words in parsing diagrams: The appearance of

the word "costs" as the main verb indicated a great deal about which sentence elements should be slotted into which equation templates.

Other features simply analyzed the syntactical relationships between words, regardless of the words themselves, while still others examined correlations between words' locations in different sentences. Finally, Kushman says, he included a few "sanity check" features, such as whether or not the solution yielded by a particular equation template was a positive integer, as is almost always the case with algebraic word problems.

"The idea of this kind of supervision they have will be useful for lots of things," says Kevin Knight, a professor of computer science of the University of Southern California. "The approach of building a generative story of how people get from text to answers is a great idea."

The system's ability to perform fairly well even when trained chiefly on raw numerical answers is "super-encouraging," Knight adds. "It needs a little help, but it can benefit from a bunch of extra data that you haven't labeled in detail."

More information: Paper: Learning to Automatically Solve Algebra Word Problems - people.csail.mit.edu/nkushman/papers/acl2014.pdf

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology