Solving Crosswords with PROVERB

Michael L. Littman, Greg A. Keim, Noam M. Shazeer
Department of Computer Science
Duke University, Durham, NC 27708-0129
{mlittman, keim, noam}@cs.duke.edu

Abstract

We attacked the problem of solving crossword puzzles by computer: Given a set of clues and a crossword grid, try to maximize the number of words correctly filled in. PROVERB, the probabilistic cruciverbalist, separates the problem into two, more familiar subproblems: candidate generation and grid filling. In candidate generation, each clue is treated as a type of query to an information retrieval system, and relevant words of the correct length are returned along with confidence scores. In grid filling, the candidate words are fit into the puzzle grid to maximize an overall confidence score using a combination of ideas from belief network inference and constraint satisfaction. For our demonstration, we will have an interactive version of the candidate-generation process available via the web, and will also give people an opportunity to go head-to-head against PROVERB in solving complete puzzles.

Crossword puzzles have been an AI staple for many years, both as an example of the constraint satisfaction paradigm (Mackworth 1977) and as a testbed for search (Ginsberg *et al.* 1990). However, we are aware of no attempts to create a broad-coverage crossword puzzle solver—one that solves crosswords based on their clues. PROVERB was developed by a group at Duke University to solve American-style crossword puzzles.

The architecture of the system consists primarily of a set of 30 "Expert Modules" responsible for suggesting solutions to the clues, and a "Solver" responsible for selecting candidate answers for each clue that fit together in the grid.

To illustrate the candidate-generation process, we took the 70 clues from the crossword puzzle published in the New York Times, Thursday, October 10th, 1998. These clues were run through the expert modules and approximately 33 were solved with high confidence (in the top 10). After grid filling (combining crossing constraints with information from the clue), 62 clues were answered correctly. We examined the 33 well-solved clues to determine which expert modules contributed to the solution. These are described below.

Modules come in several different types:

- *Word list modules* ignore their clues and return all words of the correct length from a dictionary.
- *CWDB-specific modules* make use of a crossword database (CWDB) of over 350,000 crossword clues with their solutions.
- *Information retrieval modules* retrieve answers from full text sources such as online encyclopedias.
- *Database modules* create domain-specific queries for focused databases of authors, songwriters, actors, etc.
- *Syntactic modules* solve fill-in-the-blank-type clues.

Exact Match

This CWDB-specific module returns all targets of the correct length associated with this clue in the CWDB. Confidence is based on a Bayesian calculation.

Of the 70 clues in the puzzle, 18 clues (25.7%) appeared before in the CWDB. Of these, 11 (15.7%) appeared with targets of the correct length. This is actually fairly low—average puzzles tend to have closer to 30% of their clues in the database. Of these, six appear with the correct answer, shown in bold:

clue	found
Cut off: `isolated`	amputate
Pal of Pooh: `tigger`	eeyore
Fruitless: `arid`	vain (2)
Corporate image: `logo`	**logo**
Highball ingredient: `rye`	**rye**, ice
Transfix: `impale`	**impale**
Tortellini topping: `marinara`	parmesan
"Rouen Cathedral" painter: `monet`	**monet**
Nothing _____: `less`	toit
Key material: `ebony`	**ebony**, ivory
Like mud: `oozy`	**oozy**

Transformations

Another CWDB-specific module learns a set of textual transformations which, when applied to clue-target pairs in the CWDB, generates other clue-target pairs in the database. When faced with a new clue, it applies all applicable transformations and returns the results, weighted based on the previous precision/recall of these transformations. For example,

from pairs of clues like Nymph pursuer: `satyr` and Nymph chaser: `satyr`, the module learns that clues with the word "pursuer" can be often be changed to "chaser" without affecting the meaning. For example:

clue	found
Bugs chaser: `elmer`	Bugs pursuer
Rushes: `hies`	Hurries
Pickle: `jam`	Predicament
Statue base: `plinth`	Statue stand

Partial Match

This module combines the vector space model of information retrieval with the data in the CWDB:

clue	found
Monk's head: `abbot`	Monk's superior
Playwright/novelist Capek: `karel`	Playwright Capek
Bad atmosphere: `miasma`	Poisonous atmosphere
The end of Plato?: `omega`	The end in Athens
TV captain: `kirk`	Spock's captain

Dijkstra Modules

The Dijkstra modules were inspired by the intuition that related words either co-occur with one another or co-occur with similar words. This suggests a measure of relatedness based on graph distance. From a selected set of text databases, the module builds a weighted directed graph on the set of all terms. For databases, we used an encyclopedia index, two thesauri, a database of wordforms and the CWDB. Example clues:

clue	path to target
Trigger, for one: `palomino`	trigger → palominos
Ace place?: `sleeve`	ace
Deadly desire: `envy`	deadly → deadlies, desire
4:00 service: `teaset`	service
Warner of Hollywood: `oland`	warner
Onetime electronics giant: `itt`	electronics, giant
Kind of coal or coat: `pea`	coal, coat
Meadowsweet: `spiraea`	meadowsweet

Movie

The Internet Movie Database (`www.imdb.com`) is an online resource with a wealth of information about all manner of movies and T.V. shows. This module looks for a number of patterns in the clue. Two clues in the example puzzle could have been answered by the movie module: Princess in Woolf's "Orlando": `sasha`, and "The Thief of Baghdad" role: `abu`.

Synonyms

Using Roget's thesaurus and the online thesaurus wordnet, PROVERB solves "synonym" type clues: Fruitless: `arid`, Now and again: `attimes`, Chop-chop: `APACE`.

Fill-in-the-Blanks

Over five percent of all clues in CWDB have a blank in them. We searched a variety of databases to find clue patterns with a missing word (music, geography, movies, literary and quotes). For example: "Heavens _____!": `above`.

Acknowledgements

PROVERB was developed by the Duke Crossword Team: Sushant Agarwal, Catherine M. Cheves, Joseph Fitzgerald, Jason Grosland, Fan Jiang, Shannon Pollard, Karl Weinmeister, and the authors. We received help and guidance from other members of the Duke Community: Michael Fulkerson, Mark Peot, Robert Duvall, Fred Horch, Siddhartha Chatterjee, Geoff Cohen, Steve Ruby, Nabil H. Mustafa, Alan Biermann, Donald Loveland, Gert Webelhuth, Robert Vila, Sam Dwarakanath, Will Portnoy, Michail Lagoudakis, Steve Majercik, Syam Gadde. Via e-mail, Will Shortz and William Tunstall-Pedoe made considerable contributions.

References

Ginsberg, M. L.; Frank, M.; Halpin, M. P.; and Torrance, M. C. 1990. Search lessons learned from crossword puzzles. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 210–215.

Mackworth, A. K. 1977. Consistency in networks of relations. *Artificial Intelligence* 8(1):99–118.

Shortz, W., ed. 1990. *American Championship Crosswords*. Fawcett Columbine.