CS 461 – ARTIFICIAL INTELLIGENCE

Term Project Proposal

Spring 2021

Group Nick (= Name of the Program)

CROSSWIND

Members

(S1) Melisa Taşpınar (CONTACT)
(S1) Ömer Ünlüsoy
(S1) Şebnem Uslu
(S1) Yiğit Gürses
(S1) Zeynep Berfin Gökalp

Description

In this project, we'll write a program called CROSSWIND. We'll use the "Python" programming language. Our program will input a 5x5 New York Times mini-puzzle (cf. https://www.nytimes.com/crosswords/game/mini) and its solution. It will then output the best matched solution solved by the AI and the accuracy resulting from the comparison with the actual solution. The process of constructing the output includes the establishment of the crossword database where the clue and target pairs exist. The AI will solve the given puzzle with the knowledge it gained from the previous puzzles (database) and offer a final solution in less than 15 minutes.

Literature

Crossword Solving Problem in American-style crosswords requires extensive knowledge of language, history, and popular culture. The authors mimic these knowledge and abilities with an enormous database and information retrieval techniques. Their solver system called Proverb is claimed to be the first broad-coverage computer system for solving crossword puzzles and to perform better than the average cruciverbalist (95% words correct over 370 puzzles). Proverb system models probabilistic representation and a modular system to find the best match. Each module generates a candidate list and the central module merges them and decides the best match. The database, called CWDB, comes from 5142 crossword puzzles including 350,000 clue-target pairs.

Keim, G., et al. PROVERB: The Probabilistic Cruciverbalist. In Proc.16th National Conference on Artificial Intelligence, pages 710–717, 1999. https://www.aaai.org/Papers/AAAI/1999/AAAI99-101.pdf

PROVERB: The Probabilistic Cruciverbalist

Greg A. Keim, Noam M. Shazeer, Michael L. Littman,
Sushant Agarwal, Catherine M. Cheves, Joseph Fitzgerald,
Jason Grosland, Fan Jiang, Shannon Pollard, Karl Weinmeister
Department of Computer Science
Duke University, Durham, NC 27708-0129
contact: {keim, noam, mlittman}@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. In our system, "expert modules" specialize in solving specific types of clues, drawing on ideas from information retrieval, database search, and machine learning. Each expert module generates a (possibly empty) candidate list for each clue, and the lists are merged together and placed into the grid by a centralized solver. We used a probabilistic representation throughout the system as a common interchange language between subsystems and to drive the search for an optimal solution. PROVERB, the complete system, averages 95.3% words correct and 98.1% letters correct in under 15 minutes per puzzle on a sample of 370 puzzles taken from the New York Times and several other puzzle sources. This corresponds to missing roughly 3 words or 4 letters on a daily 15 × 15 puzzle, making PROVERB a better-than-average cruciverbalist (crossword solver).

Introduction

Proverbs 022:021 *That I might make thee know the certainty of the words of truth...*

Crossword puzzles are attempted daily by millions of people, and require of the solver both an extensive knowledge of language, history and popular culture, and a search over possible answers to find a set that fits in the grid. This dual task, of answering natural language questions requiring shallow, broad knowledge, and of searching for an optimal set of answers for the grid, makes these puzzles an interesting challenge for artificial intelligence. In this paper, we describe PROVERB, the first broad-coverage computer system for solving crossword puzzles[1]. While PROVERB's performance is well below that of human champions, it exceeds that of casual human solvers, averaging over 95% words correct over a test set of 370 puzzles.

We will first describe the problem and some of the insights we gained from studying a large database of crossword puzzles; these motivated our design choices. We will then discuss our underlying probabilistic model and the architecture of PROVERB, including how answers to clues are suggested by expert modules, and how PROVERB searches for an optimal fit of these possible answers into the grid. Finally, we will present the system's performance on a test suite of daily crossword puzzles and on 1998 tournament puzzles.

The Crossword Solving Problem

The solution to a crossword puzzle is a set of interlocking words (*targets*) written across and down a square grid. The solver is presented with an empty grid and a set of clues; each clue suggests its corresponding target. Some clue-target pairs are relatively direct: ≺Florida fruit [6]: `orange`≻[2], while others are more oblique and based on word play: ≺Where to get a date [4]: `palm`≻. Clues are between one and a dozen or so words long, averaging about 2.5 words per clue.

To solve a crossword puzzle by computer, we assume that we have both the grid and the clues in machine readable form, ignoring the special formatting and unusual marks that sometimes appear in crosswords. The *crossword solving problem* is the task of returning a grid of letters, given the numbered clues and a labeled grid.

In this work, we focus on American-style crosswords, as opposed to British-style or cryptic crosswords. By convention, all targets are at least 3 letters in length and long targets can be constructed by stringing multiple words together: ≺Don't say another word [13]: `buttonyourlip`≻. Each empty square in the grid must be part of a down target and an across target.

As this is largely a new problem domain, distinct from crossword-puzzle creation (Ginsberg *et al.* 1990), we wondered how hard crossword solving really was. To

[1] Crossword Maestro is a commercial solver for British-style crosswords published by Genius 2000 Software. It is intended as a solving aid, and while it appears quite good at thesaurus-type clues, in informal tests it did poorly at grid filling (under 5% words correct).

[2] Target appears in fixed-width font; all examples are taken from our crossword database (the CWDB). We will note the target length following sample clues in this paper to indicate a complete specification of the clue.