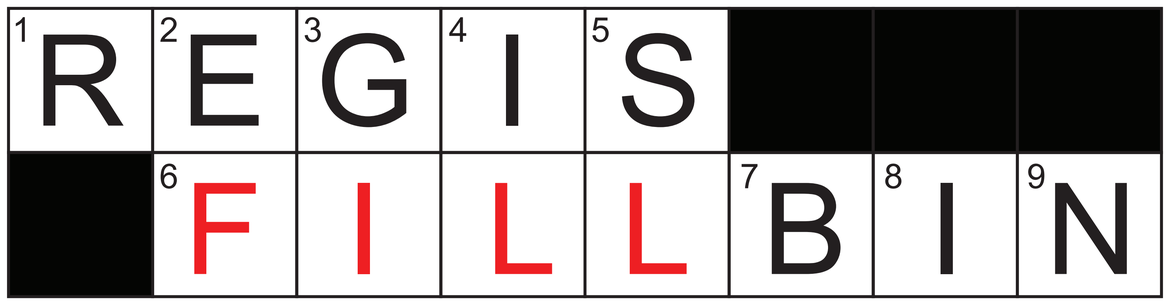


Sam Hage

Introduction

Crossword puzzles present several interesting problems from a computer science perspective, because they comprise a few processes at which humans remain superior to computers. My program aims to surmount these problems and solve a New York Times crossword. It takes its name from Dr. Matt Ginsberg's 2011 paper and corresponding program, "Dr.Fill". The program works by generating lists of candidate answers for each clue, then searching for the configuration of answers that best satisfies all constraints (constraints being the intersecting answers for other clues).

The New York Times crossword was chosen as the focus of the project because it is the best written and edited crossword, by general consensus among avid solvers. Consequently, the best crossword solving computer program should be able to solve the New York Times puzzle.

```
('frying fat', 'LARD')
('frying liquid', 'CORNOIL')
('frying liquid', 'OIL')
('frying medium', 'CORNOIL')
('frying medium', 'DEEPFAT')
('frying medium', 'LARD')
('frying mess', 'SPATTER')
('frying need', 'FAT')
('frying pan hazard', 'SPATTER')
('frying pan mishap', 'SPATTER')
('frying pan sound', 'SSS')
('frying pan spray', 'PAM')
('frying pan', 'GRIDDLE')
('frying pan.', 'SPIDER')
('frying pans.', 'SPIDERS')
('frying sound', 'SSS')
('frying, for one', 'PAN')
('frying-butter sound', 'SSS')
('frying-pan coating', 'TEFLON')
('fryolator fill', 'DEEPFAT')
('fsu and nc state are in it', 'ACC')
('fsu or uf', 'SCH')
('ft. --- (former military base near monterey, ca)', 'ORD')
('ft. ---: former california military base, near monterey', 'ORD')
('ft. above land', 'ALT')
```

A screenshot of the data used in solving the puzzles

```
3       10      down    ['units named for physicist enrico', '??????']  ['FERMIS']
4       13      down    ['land chronicled by c. s. lewis', '??????']    ['NARNIA']
4       14      down    ["grammar nazis' concerns", '??????']   ['USAGES']
5       6       down    ['greeting in rio', '???']      ['OLA']
5       11      down    ["skirt's edge", '???'] ['HEM']
7       8       down    ['six-sided roller', '???']     ['DIE']
10      5       down    ['dupont acrylic fiber', '?????']               ['ORLON']
10      10      down    ['kind of energy with panels', '?????'] ['SOLAR']
12      1       down    ['poem of praise', '???']       ['ODE']
0       11      down    ['letter after x-ray and yankee in the nato alphabet', '????']  ['ECHO', 'ZULU']
7       3       down    ['floored, as a boxer', '???']  ['MAS', 'ALI']
11      12      down    ["slight hitch in one's plans", '????'] ['REUP', 'SNAG']
1       6       across  ['norse deity with a hammer', '????']   ['THOR', 'ODIN', 'IDUN']
0       9       down    ['nudges', '?????']     ['PRODS', 'PESTS', 'POKES']
11      3       down    ['vases', '????']       ['MING', 'URNS', 'OILS']
0       1       down    ['by any other name it would smell as sweet, per juliet', '????']       ['ROSE', 'NAME',
'ARAB', 'IKEA']
0       6       across  ['heed a red light', '????']    ['STOP', 'EMTS', 'PINK', 'IDLE', 'CORN', 'EXIT']
8       12      across  ['180 degree turn, informally', '???']  ['SSE', 'ESE', 'ENE', 'UEY', 'DOA', 'ELL']
10      0       across  ['three-time foe for frazier', '???']   ['ORC', 'ALI', 'FOE', 'RAF', 'NAT', 'NEO']
13      5       across  ['store sign between 9 a.m. and 6 p.m.', '????']        ['AGUN', 'ARGO', 'ALMA', 'SIMS',
'RHEE', 'MOON', 'SOSA', 'ABMT']
12      0       down    ['pre-airconditioning cooler', '???']   ['ITD', 'ICE', 'ACS', 'ADE', 'CON', 'FAN', 'PEN',
'TEA', 'EIS']
6       4       down    ['fish that can attach itself to a boat', '??????']     ['FINGER', 'PLAQUE', 'REMORA',
'ELOISE', 'OTTAWA', 'AEGEAN', 'BODEGA', 'PEELER', 'CLEAVE', 'BUCKLE']
9       0       across  ['"i\'ll handle it!"', '?????'] ['ONONE', 'HITME', 'LETME', 'WIELD', 'EWERS', 'PAWED', 'NAPES',
'IDTAG', 'IDAHO', 'TEXAS', 'WAGON', 'ALIAS']
```

Lists of answers generated by the three search modules

```
## MONDAY SEPTEMBER 5 2015 ##
# ACROSS
0       0       5       top of a wave
0       6       4       heed a red light
0       11      4       tanginess
1       0       5       do-it-yourselfer's book genre
1       6       4       norse deity with a hammer
1       11      4       part of the eye
2       0       5       chris who sang "Wicked Game," 1991
2       6       4       guthrie of Rising Son Records
2       11      4       word repeated before "pants on fire!"
3       0       13      showtime series named after an old fiction genre
4       3       3       proverbial madhouse
4       7       4       "when all ___ fails, read the instructions"
4       12      3       young-sounding wildebeest
5       0       4       spydom's ___ hari
5       5       4       ____-cola
5       10      5       cousins of ostriches
6       0       3       early afternoon hour
6       4       4       cheese off
```

Sample plaintext puzzle input

Finding Answers

Much of the work on this project was done in generating the candidate answer lists—the methods and data used will be described in this section.

Regis Fillbin's primary data source is a corpus of single clue-answer pairs drawn from nearly twenty years of previous puzzles. It performs an initial search on this data, and another on a reduced version of the same data. The program also draws on a word list generated from a crossword dicitonary, and a parsed list of all Wikipedia article titles.

The first search module is a binary search on the alphabetically sorted list of clues and answers. Rather than using exact string matching, this search incorporates the Levenshtein distance to judge whether two clues have the same meaning. The theory behind this type of search is that similar clues arise across many crossword puzzles. Certain clues, therefore, even from a brand new puzzle, will show up in or be a very close match for a clue in the data set. Experimentally, this search will generate a match for one-third to one-fourth of a puzzle's clues. The accuracy of these answers is classified with a high degree of confidence.

Next a keyword-based search using Python's regex library is performed to generate answers for the rest of the clues. This module does not have the benefit of logarithmic performance, as the binary search does, and ends up much slower. Consequently, before performing the search, the data is compressed by randomly discarding pairs from the set. This leaves approximately one-third of the original data and results in much faster performance.

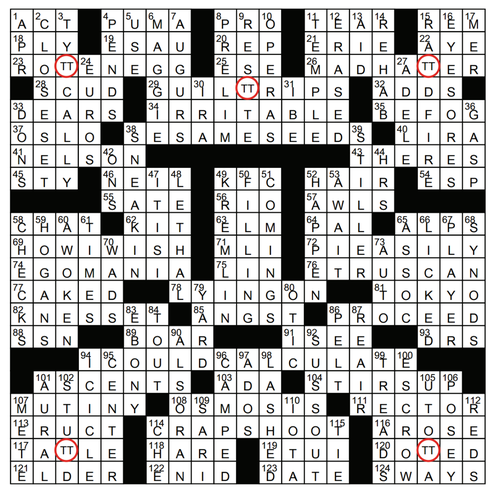
The third and final module is performed repeatedly, after the first two. It takes all clues whose answers still have blank squares and runs these patterns of known and unknown letters against the two dictionaries. This module generates even more candidates, and therefore more false positives, since it ignores the clue entirely and focuses purely on the letters.

Filling the Puzzle

Regis Fillbin's approach to filling the puzzle is similar to the way a human would solve a crossword; it begins with the answers in which it is the most confident, moving to less confident answers, updating other answer lists by eliminating candidates that would conflict with crossing answers. Although, strictly speaking, the program uses unweighted constraint satisfaction (the individual candidates are not weighted within the lists), in effect it does contain an element of weighting. Answers gathered from the first module are treated with greater confidence because of their closeness to previously seen clues. What's more, the candidates from shorter answer lists are incidentally weighted higher, because there are simply fewer possibilities.

Repeated tests on Monday and Tuesday puzzles have shown good results, generally over 75 percent of squares filled correctly. I found this performance acceptable for a prototype, but actually worse than most humans.

Because of random data compression, one could expect highly variable performance between trials. While some variance is evident, the solutions tend to be similar, likely because of a high degree of redundancy in the full data set.



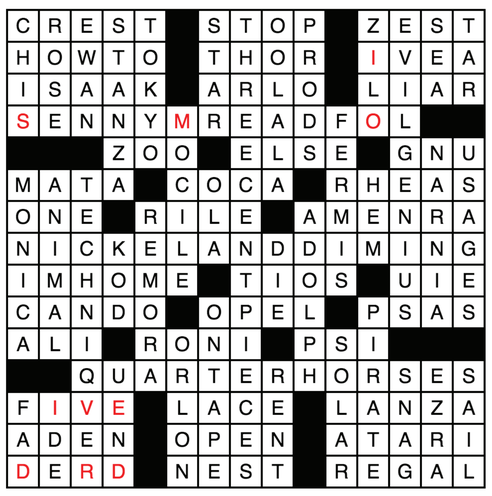
Three examples of rebus puzzles

Limitations and Future Work

Many crosswords have themes; related longer answers that appear throughout the grid and are generally tied together with a single "theme clue". These themes almost always involve wordplay, intentional misspelling, or other humor; famously difficult problems for computers. Regis Fillbin's natural language processing algorithms will fail to correctly interperet such wordplay.

Richtext, non-alphanumeric characters, and so-called "rebus" puzzles also exceed the program's capabilities. These puzzles will have a special character, multiple characters, or even an entire word in a single square, generally requiring some level of human intelligence to interperet them correctly. Other puzzles will have a shape described in the black squares of the grid itself.

Clearly, there is room to expand Regis Fillbin's capabilities in these areas, but the biggest focus of the next semester will be resolving the current tradeoff between speed and accuracy. Because of the random data compression, vital data may simply be discarded on a given trial. With- out compressing the data in some way, however, the run time is too long to be truly practical (around 15 minutes compared to 4). The easiest way to resolve this will be partitioning the large data into subsets based on answer length, allowing the second search module to run on a smaller data set with- out actually discarding any relevant data. There is also room to improve the program's performance by incorporating more natural language processing and less data analysis.



Two puzzles solved by my program, a Monday (left) and a Tuesday (right). Incorrect letters are in red.