A Play on Words

Cognitive Computing as a Basis for AI Solvers in Word Puzzles

Thomas Manzini, Simon Ellis, Prof. Jim Hendler

Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, 12180

Introduction

Word puzzles are a fun pastime for people of all ages. They range in difficulty from simple word searches to cryptic crosswords and can leave many human players scratching their heads. One such type of word puzzle is the *syllacrostic*. This type of puzzle gives the player a series of clues and a list of syllables from which to chose. Each clue has a number of letters and syllables to which the answer must conform. The player must solve the puzzle by identifying the proper word, with the right number of letters and syllables, for each clue. A brute force solver creates several million possible answers that must be reviewed to identify the best response. The aim of this project was to develop a cognitive computing system that is able to use knowledge and reasoning to select fewer candidates and to use scorers and a ranking system to determine which are relevant and correct. Figure 1 below depicts an example of a syllacrostic game.

Syllable List

AN AP BLE CAR CES DEN DER DRIV ER FRIC GAR HU IM IST KIN MOR NA NENT
NI OR PA PER POS PREG RA RI SA SI TE TEN TI TILE TION TRY TUS TY VAL VER

1. Heat generating force _____ (2)	(Friction)
2. Traveling show _____ (3)	(Carnival)
3. Relatives _____ (3)	(Ancestry)
4. Solidity _____ (3)	(Density)
5. Golf club _____ (2)	(Driver)
6. Comedian _____ (3)	(Humorist)
7. Invincible _____ (4)	(Impregnable)
8. School for young children _____ (4)	(Kindergarten)
9. Rear _____ (4)	(Posterior)
10. Equipment _____ (4)	(Apparatus)
11. Having Many Skills _____ (3)	(Versatile)
12. Relates to the topic _____ (3)	(Pertinent)

Figure 1 Example Syllacrostic Game
Source: http://www.aviewofamerica.com/

Example

Each game has a clue (often a single word or concept), a series of blank spaces indicating letters, and a number at the end indicating the number of syllables for the answer. The game shown above is a simple example of a syllacrostic, but even in such a simple game, a brute force solution for Clue 8 (with 12 letters and 4 syllables) would generate 1,771,560 candidate answers, the majority of which would be nonsensical. This approach is incredibly time inefficient and the number of candidate answers can grow very quickly depending upon the type of clue and game that is being solved.

In contrast, our cognitive computing solution chooses only 767 candidate answers for Clue 8 and successfully selects the correct answer from this list. Following the removal of all the answers that do not meet the proper letter or syllable count and cannot be constructed using the list of available syllables, there are fewer choices for the final word. This leaves room for further analysis of the candidate answers to determine which of the remaining words is the best response.

Acknowledgements

Special thanks to the project supervisor, Simon Ellis, for all of the wonderful support and advice that he has given throughout this project. The same to the project's research advisor, Dr. Jim Hendler, for all the encouragement, direction, and drive that he has given. Thanks is also extended to the Department of Computer Science for their help as well.

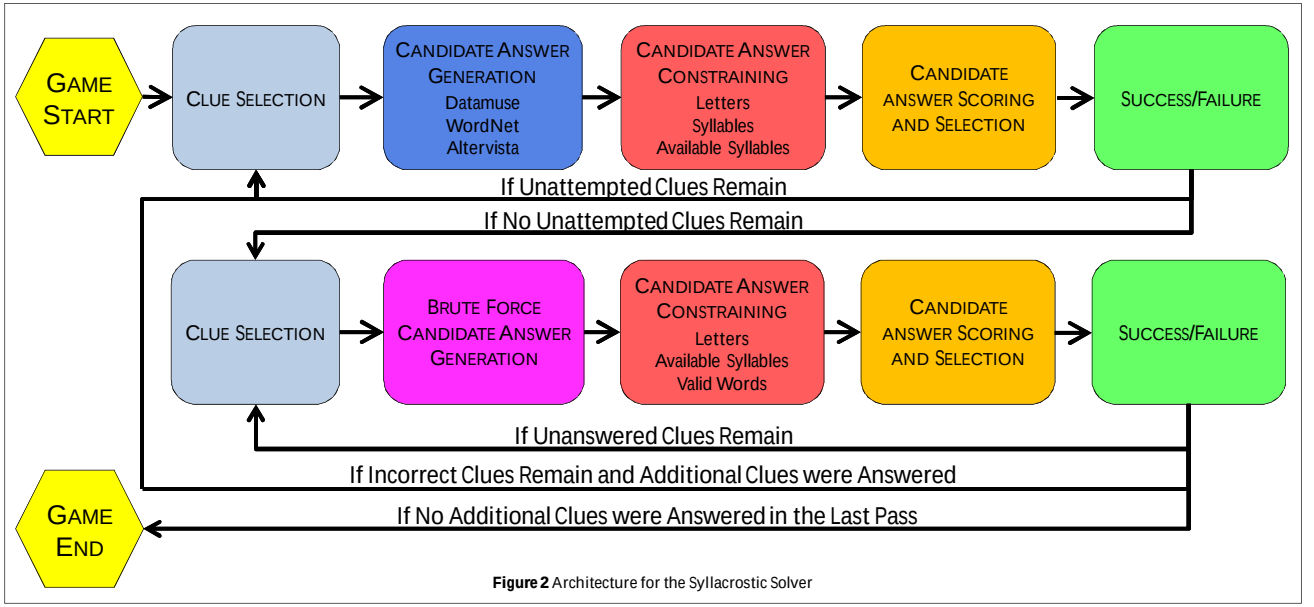

Figure 2 Architecture for the Syllacrostic Solver

Method

Our approach uses a candidate answer generation process, followed by scoring, as described in Figure 2. This approach is inspired by the Watson system, developed by IBM, and the approaches that human players take when solving these puzzles.

The process generates many different candidate answers for each clue using a variety of data sources (Datamuse, Altervista, WordNet), and then constrains them based on the number of letters and syllables and the available syllables remaining. The remaining words are then scored by several different measures, including part of speech, definition, and the number of times that different sources generated the candidate. Following the scoring, the candidates are ranked and the candidate with the highest rank is selected.

Once the initial process is complete for all clues, our system takes all of the clues for which it was unable to find a solution and attempts to brute force a response by creating valid words from the list of unused syllables. It then constrains these brute forced words by the clue limits (letter count and syllable count) and then scores the remaining words and selects a solution. While a brute force solution is typically something to be avoided, it is a valid approach because it is a process that humans take to solve these puzzles.

All clues for which the solution was incorrect have their solutions erased and the process is repeated, with additional information in the form of eliminated syllables. This is repeated until a pass is been made in which no additional clues have been solved.

Currently, the system can solve syllacrostic clues with roughly 97% accuracy. The average time needed to solve a clue is roughly one second.

Scoring

The system uses several different scorers to determine the score of a candidate. A part of speech scorer attempts to match the part of speech that is expected as an answer. An additional score of word similarity is computed using WordNet and is based on the maximum similarity between any word in the clue and the candidate answer. A relevancy score is automatically returned from the Datamuse API. An additional scorer is used for tracking whether a candidate was generated using the Altervista Thesaurus. Finally, an additional scorer computes the number of instances in which the same candidate was generated from different sources. All of these scores are weighted equally and the candidate with the highest sum of scores is ranked highest.

Future work

- Develop and train machine learning classifiers to rank the candidate answers.
- Add additional sources for candidate answer generation.
- Create more accurate and more effective scorers.
- Expand the system to crosswords and other word puzzles.
- Add additional support for more abstract clues (e.g. “bring on the germs” → “Contaminate”).