CS221 Project: Scrabble AI

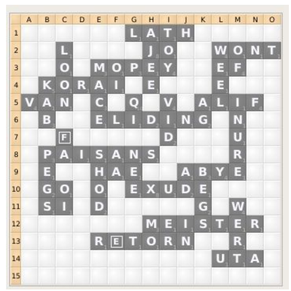
Colleen Josephson, Rebecca Greene
Department of Electrical Engineering, Stanford University

Motivation

- Stochastically partially observable game
- Incredibly large state space
  - Standard decision tree models don't work

= Interesting problem!

Three Subproblems



Search Problem
- Combine 7 tiles with anything on the board
- Dictionary of ~200,000 possible words
- Need to find possible valid moves

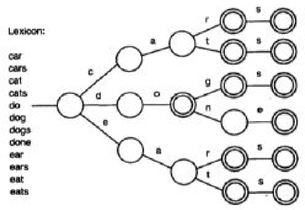
Move Selection
- Good players willing to sacrifice points on current turn to increase probability of high-scoring future moves

Opponent Modeling:
- Avoid creating opportunities for opponent to make high-scoring moves

Challenges

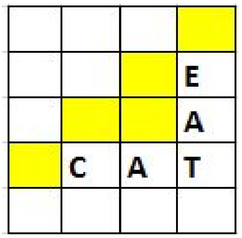
- Combining many AI techniques: backtracking search, Monte Carlo, machine learning
- Quackle vs. cs221 autoplay
  - Quackle is large C++ codebase
  - Interface needed between their C++ and our Python

Approach: Search



1. Model the lexicon as a trie

2. Define starting 'anchor points'




```
There are 191 legal moves:
[('JO', (6, 8), 'v', 17), ...
('RAJES', (4, 10),'v', 36),
('TRAPS', (7, 9), 'v', 9)])
max word RAJES with score 36
```

3.Solve using a modified backtracking search

Approach: Move Selection

- After generating list of possible moves, pick best move as weighted sum of score and rack management heuristics
- Features Include: letters, doubles, triples, vowel/consonant ratio, 'qu'
  - 'qu' = whether or not 'q' and 'u' are in the same rack
- Use machine learning & stochastic gradient descent to find weights

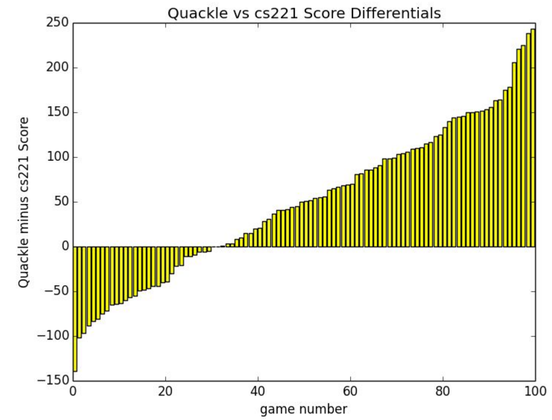
$$Val(\text{word}, \text{rack\_remains}) = a score(\text{word}) + \beta(\#vow/\#cons) + \gamma(A \in \text{rack\_remains}) + \ldots$$

Approach: Opponent Modeling

- Replace 'score' with probabilistic score differential from depth-2 Monte Carlo simulations
- Use observed tiles and moves to infer probability of different opponent's racks
- Weight score differentials accordingly

Preliminary Data

Metric: play our AI vs. Quackle, an open-source AI that has defeated human Scrabble champions.



NOTE: We limited Quackle's moves to 20s due to deadline constraints, so this is an optimistic dataset.

Analysis

- Score differential for 100 games
  - Average differential is 50 points
  - Quackle average score 297
  - cs221 average score of 247
- Our AI generally does worse, but wins 29%
  - Limited to Quackle to 20s per move, so not performing at full capacity
    - 5 minutes and unlimited also available
  - Scrabble is probabilistic, luck of the letter draw is a major factor
- Quackle wins by larger margins
  - 2x or more points 5% of the time
    - Max cs221 win is factor of 1.6
  - Completed Monte Carlo will hopefully narrow gap

Remaining Work

- Gather data for non-time limited case
- Gather data for Monte Carlo enabled play
- Ensure Quackle runs on corn.stanford.edu
- Make software easy to run and clearly documented