

Reproducing A Competitive Scrabble Agent

Casey Pore

Colorado State University
cpore@rams.colostate.edu

Problem Description

What were you trying to do; this should describe the problem addressed by the project. Use your original proposals as a starting point for this part. Also, include your hypothesis(es) here along with a precis of what you found.

In this paper, I describe the process of building and evaluating a Scrabble agent from scratch that is capable of competitive play. The unique properties of Scrabble make it difficult to apply existing generalized AI strategies common to other games, such as an adversarial search. In fact, even the best Scrabble agent (Brian Sheppard's Maven) do not utilize any local search techniques until the very end of the game, although it has been shown that the end-game search can be crucial for cinching a win against the best human players (Sheppard 2002). As we will see, creating a competitive Scrabble player is more about adding features that incrementally improve the agent's chances for higher scoring, such as a combination of move selection heuristics and end-game search, than it is any one over-arching concept of Artificial Intelligence. This sentiment was expressed by Brian Sheppard when he said his program, Maven, "is a good example of the 'fundamental engineering' approach to intelligent behavior" (Sheppard 2002).

Scrabble is a game of imperfect information. Outcomes are highly dependent on the usefulness of the tiles a player randomly draws from the bag to make high scoring moves. Additionally, it cannot be known what tiles the opponent holds. This makes it difficult (or at least futile) for an agent to plan ahead or anticipate future states of the game from which to make informed decisions about how to play its tiles. This limits our options for quickly generating moves an agent may select to play. To accomplish the goal of generating possible move quickly, an efficient data structure along with an algorithm for identifying valid moves lie at the heart of my implementation, as well as agents that have come before mine. Once this fundamental technique was in place, I worked toward adding heuristics to improve the chances of maximizing the average move score in hopes of drawing my agent closer to being as competitive a player as Maven.

After completing the move generation algorithm, the first heuristic implemented was to simply choose the move which

produced the the highest score. I found that my implementation, though slightly slower than Appel and Jacobson's player from which I based my implementation, outperformed their player in regards to average move score by a significant margin (Appel and Jacobson 1988). Next, I implemented several heuristics (taking inspiration from a few different sources) under the hypothesis that each heuristic would add a statistically significant increase to the average move score of my agent and increase its win ratio. An agent was created for each of these heuristics and pitted against an agent using only the basic maximum score heuristic to evaluate their effectiveness. Finally, after evaluating which of these individual heuristics led to a better outcome, I implemented several agents that use different combinations of the heuristics hypothesizing that they would lead to even better outcomes. These multi-heuristic agents were pitted against the maximum score agent to evaluate which combinations worked the best. I found that almost all of the multi-heuristic agents played better than any single-heuristic agent for creating strong Scrabble agent, however, not all of the single-heuristic agents bested the maximum score agent.

Previous Work

survey what approaches have been adopted before. It may be a quick reference to the AI techniques selected for the project or it could reference papers on the problem or on the previous approaches taken to the problem. It depends on the composition of the project; if it extends previous work, then reference what it extends; if it is novel, then reference techniques that motivated the design. References should be from publications rather than web sites; the exception to that is if you downloaded code or problems from a website.

Approach Taken

what did you do. Describe your project: what AI techniques are used by it, why you picked these techniques, how was the project structured, who did what, what sort of data was supplied (example problems for learning systems, prior models for non-learners), what results were expected. Code samples must be short.

Evaluation and Analysis

How well did your project do: as expected, better or worse. Why did it perform as it did? What worked and what did not? Were there any surprises? What experiments/evaluation did you run? Include your experiment design to this section; it may have a subsection for the experiment design and another for the analysis.

Future Work, Conclusions

What did you learn? What would you do in addition or differently? (Torre and Verducci 2008) (Young 2013) (Shapiro and Smith 1982)

References

- Appel, A. W., and Jacobson, G. J. 1988. The world's fastest scrabble program. *Commun. ACM* 31(5):572–578.
- Shapiro, S. C., and Smith, H. R. 1982. A scrabble crossword game playing program. Technical Report 119, Indiana University.
- Sheppard, B. 2002. World-championship-caliber scrabble. *Artificial Intelligence* 134(1):241–275.
- Torre, J., and Verducci, T. 2008. *The Yankee Years*. Doubleday.
- Young, S. 2013. Building a program that plays scrabble. <http://www.scotthyoung.com/blog/2013/02/21/wordsmith/>.