Annotated Bibliographies

Pitch 1: Euchre AI

My plan is to create an AI to play the card game Euchre. I've played with a number of computer partners/opponents before, but have found them to be unsatisfactory. They make moves that seem illogical, and end up performing poorly. Some research has been done regarding the game, but it is largely outdated. Thus, I plan to use some of the more recent techniques in machine learning in order to find what works well. Particularly, Monte Carlo Search Trees and Reinforcement Learning seem to be promising methods.

Bibliography

- Cooperative Artificial Intelligence in the Euchre Card Game
- Seelbinder, Benjamin E. 2012. *Cooperative Artificial Intelligence in the Euchre Card Game.* Master's thesis. University of Nevada, Reno.
- [PDF] unr.edu

This thesis provides a solid foundation of terms and earlier techniques that have been used in the field of Euchre AI. The simplest of models use a straight forward high/low strategy to try to maximize individual wins. However, Euchre is a cooperative game played with a partner, so Seelbinder proposes methods that incorporate this aspect more. Further, as Euchre is an imperfect information game, he uses Markov Decision Processes to factor in probabilities. This paper does not get into more modern techniques in machine learning, so its strengths lie more in digging into the strategies of the card game.

- Learning to Play Euchre With Model-Free Reinforcement Learning
- Pugh, Eli. Learning to Play Euchre With Model-Free Reinforcement Learning.
- https://web.stanford.edu/class/aa228/reports/2020/final165.pdf

This article is a significantly more up-to-date work in the field of Euchre AI. Specifically, it dives into the methods of Deep Q Neural Networks and Neural Fictitious Self Play. According to their results, both of these methods have potential at rivaling agents created using a human-like, rule-informed method. They use an open source toolkit called RLCard to run their experiments which might potentially be a good tool to work with. In the discussion, the article references Monte Carlo Tree Search as a potentially good algorithm for further investigation in the field.

- Information Set Monte Carlo Tree Search
- Cowling, P. I., Powley, E. J., & Whitehouse, D. 2012. Information set monte carlo tree search. *IEEE Transactions on Computational Intelligence and AI in Games*, *4*(2), 120-143.
- [PDF] whiterose.ac.uk

This paper deals with applying the ideas of Monte Carlo Search Trees (MCST) to games of imperfect information. To do this, they first describe MCST and identify a few different types of imperfect information games. Euchre fits into the category of having imperfect information sets, as each player's cards are not revealed to the others. The algorithms introduced utilize determination. This method pretends that it is in a number of perfect information games and does rounds of sampling based on those to decide what the best move might be. The paper does not factor belief distributions based on playing history into the mix. Considering the large amount of guidance that can be gained from this, however, I believe it is something I will want to consider.

- Towards computationally efficient responsibility attribution in decentralized partially observable MDPs
- Triantafyllou, S., & Radanovic, G. 2023. Towards computationally efficient responsibility attribution in decentralized partially observable MDPs. *arXiv preprint arXiv:2302.12676*.
- [PDF] arxiv.org

In this paper, the ideas of MDPs, responsibility attribution, and MCSTs are discussed and then evaluated in three partner based card games. It notably dives deeply into the technical and mathematical analysis of the approach. The work is largely concerned with causality and how to correctly attribute that to the players and the actions that take place. In the Euchre domain, they reference the HIGH! agent from Seelbinder's model.

- Rlcard: A toolkit for reinforcement learning in card games.
- Zha, D., Lai, K. H., Cao, Y., Huang, S., Wei, R., Guo, J., & Hu, X. 2019. Rlcard: A toolkit for reinforcement learning in card games. *arXiv preprint arXiv:1910.04376*.
- [PDF] arxiv.org

This paper provides an introduction to the environment in which Pugh’s experiments were conducted. It is a toolkit designed to be used for running reinforcement learning techniques on card games. The paper describes a number of challenges in large, hidden information card games, and then outlines the basics of the program’s game structure and ways of handling them.

- Over 1 billion tricks played - Information Set Monte Carlo Tree Search Euchre simulation database
- Bravender, Dan. 2017. Over 1 billion tricks played - Information Set Monte Carlo Tree Search Euchre simulation database
- https://dan.bravender.net/2017/6/9/Over_1_billion_tricks_played_-_Information_Set_Monte_Carlo_Tree_Search_Euchre_simulation_database.html

This is an informal post as opposed to a published article, so it is not especially well-written, clear, or thorough, but it does directly apply Monte Carlo Tree Search to the problem of Euchre. Bravender finds that his simulation supports many commonly held ideas about the strategies of

Euchre such as the benefit of always leading the right bower if you have it. Additionally, he links to his gitHub repository which might be good to use or look at.

Pitch 2: Japanese Translation
This project would look to see what can be done of translating basic Japanese sentences using search techniques. Google translate is still very bad (from my personal experience) at translating Japanese to English, so I’d be curious to see what technique might lead to better translations. The first step in machine translation is to split a sentence up into “words”. As Japanese does not contain spaces to separate words, this task is significant. For example、the sentence: わたしのともだちはいちごがすきです。might be split into the tokens: ‘わたしの’、’ともだちは’、’いちごが’, and ’すきです’. But, there are still a lot of options for translation. If we take the list in direct sentence order and use the best word-for-word translation, we end up with “My friend strawberry like”, but the preferable English sentence would be “My friend likes strawberries”. This is where a search algorithm will help us look through the possibilities. Beam search is a good candidate as it keeps a number of possible translations at each step, but not enough to make it computationally impractical in this large search space. Additionally, we probably can use information from grammatical structures like the particles ‘の’、’は’、and ’が’ as part of the heuristic that guides us.

Bibliography

- Optimal beam search for machine translation
- Rush, A. M., Chang, Y. W., & Collins, M. 2013. Optimal beam search for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 210-221).
- [PDF] aclanthology.org

This paper looks at the use of beam-search in the field of machine translation. Specifically, it is concerned with guaranteeing that the solution returned is optimal and performance is quick. They claim that their method works well for both syntax-based translation and phrase-based translation, though as all examples given use phrase-based translation models, it would be good to check out their references for syntax-based models as well. I am interested in ensuring better translations, so having methods to guarantee optimal solutions is a promising field of work.

- Pharaoh: a beam search decoder for phrase-based statistical machine translation models
- Koehn, P. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Machine Translation: From Real Users to Research: 6th Conference of the Association for Machine Translation in the Americas, AMTA 2004, Washington, DC, USA, September 28-October 2, 2004. Proceedings 6* (pp. 115-124). Springer Berlin Heidelberg.
- [PDF] academia.edu

This article follows an actual application of this sort of model for machine translation. It covers not just an overview of how the field works, but dives deeply into actually applying the theory.

Notably the paper spends significant time discussing ideas for choosing which hypothesis to prune. In order to make something that translates efficiently, how to handle beam pruning will be a large concern. The decoder described in the paper, Pharaoh, is free and thus might be worth examining in the future.

- An Efficient A* Search Algorithm for Statistical Machine Translation
- Och, F. J., Ueffing, N., & Ney, H. 2001. An efficient A* search algorithm for statistical machine translation. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*.
- [PDF] aclanthology.org

This article takes a look at comparing the popular search algorithm A* with beam search in the domain of machine translation. During these experiments it became clear that having a solid heuristic is very influential on the running of A*. Though Beam generally runs faster and takes less computational power to return good solutions, A* ensures optimal solutions. Thus, if I am trying to implement a variant of Beam to complete translation tasks, it might be worth comparing to A* as a benchmark in its development.

- An Interactive Japanese Parser for Machine Translation
- Maruyama, Hiroshi.1990. An interactive Japanese parser for machine translation. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics.*
- [PDF] aclanthology.org

This paper takes a look at the structure of Japanese sentences and covers a bit about how a parse tree containing the different components might be constructed. Their method is a system for guided parsing with the help of non-expert native speakers, but will also run without user guidance. The paper is useful for introducing the more linguistic side of translation and connecting that to the computer science side as it looks into word dependencies and modifiers. Additionally, the author also has other works in the field that may be worth checking out as well.

- A framework of a mechanical translation between Japanese and English by analogy principle
- Nagao, M. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. *Artificial and human intelligence*, 351-354.
- [PDF] aclanthology.org

This well known, older work in the field also provides a stronger connection between the fields of Japanese linguistics and machine translation. Particularly, it points out that due to the difference in Japanese and English sentence structure, many of the methods developed for other translations work poorly. The article, using clear examples, outlines a different approach that’s based on more organic human translating methods like using analogies and word association developed from dictionaries and thesauri. That is, taking a larger picture look at the meanings of sentences and relationships between words in the sentences to inform the way the sentence will be translated.

- Backward beam search algorithm for dependency analysis of Japanese
- Sekine, Satoshi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2000. Backward beam search algorithm for dependency analysis of Japanese. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.
- [PDF] aclanthology.org

This paper discusses the idea of dependency analysis and proposes the use of beam search to help complete this task. Particularly, the search looks through the sentence backwards as they have found this to be a more efficient way to analyze Japanese sentences. An interesting problem they point out is that having a higher beam width does not necessarily lead to better results. They use the Kyoto University Text Corpus data set, which is likely a promising resource.