Opponent Modeling in Automated Bilateral Negotiation

Can Machine Learning Techniques Outperform State-of-the-Art Heuristic Techniques?

Tudor Octavian Pocola
Supervisors: Pradeep K. Murukannaia, Bram M. Renting
EEMCS, Delft University of Technology, The Netherlands
23-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Automated negotiation agents can highly benefit from learning their opponent's preferences. Multiple algorithms have been developed with the two main categories being: heuristic techniques and machine learning techniques. Historically, heuristic techniques have dominated the field, but with the recent development in the field of machine learning, this is no longer true. The main goal of the paper is to compare these two techniques quantitatively using the Pearson correlation of bids. The models that were chosen as the heuristic and machine learning baseline are the Smith and the Perceptron models, respectively. Our results show that the two baselines have similar performance. This leads us to conclude that machine learning algorithms have caught up with their heuristic counterparts. Furthermore, we have also found a statistically significant correlation between the Perceptron model's accuracy and the seen bid space.

1 Introduction

Negotiation has shaped our civilization from the dawn of time, but, counter-intuitively, humans appear to be ill-equipped for negotiation [7]. An investigation conducted on senior-level executives found that "95 percent reached suboptimal outcomes in a realistic business simulation" [16].

Automated negotiation [5] offers a possible solution to this problem. The application of this system includes: negotiating the right of passage between pedestrians and self-driving cars [8] or negotiating prices on an intelligent energy grid [6].

One of the main issues the negotiation agents face is not having access to their opponent's preference profile, which is usually kept private. However, learning the opponent's preferences can significantly improve the effectiveness of automated negotiation programs [3; 4; 13], so it is an actively researched topic in the field of automated negotiation.

Several techniques have been developed for estimating the opponent's preferences [9; 17; 18; 19], each having its strengths and weaknesses. Two main types of opponent modeling techniques can be identified: heuristic algorithms and machine learning.

The heuristic methods have been researched extensively [9; 17]and the state-of-the-art (SOTA) models have been identified [4]. A 2013 comparison [4] found that most SOTA heuristic techniques converge quickly towards their maximum achievable accuracy, but they tend to decay towards the end of the negotiation when the opponents change their strategy. Furthermore, the same study concluded that most SOTA algorithms are close to perfect accuracy, so the field has little room for improvement.

Machine learning has historically been dominated by its heuristic counterparts when used to model the opponent [4]. However, this has started to change in recent years [19] due to new improvements in machine learning techniques. These new developments have also been extended to calculate the opponent's preferences [10].

Given the improvements that machine learning has seen in recent years, it is not clear which of the two methods presented above is better at estimating the opponent's preference profile. Furthermore, machine learning techniques have surpassed their heuristic counterparts in other fields, with a noteworthy example being the recent victory of AlphaZero [15] over Stock Fish 8 in the game of chess. However, no recent study that was concerned with automated negotiation has compared these two techniques directly, so a scientific gap is present.

The goal of this paper is to fill this gap by answering the following research question: *How do machine learning techniques compare with the SOTA heuristic techniques when used to calculate the opponent's preferences?*

2 Related Work

The first type of related work contains literature studies [1; 3; 13], which analyze multiple opponent modeling techniques from a qualitative perspective. They do not compare the different techniques directly but offer the strengths and weaknesses of each method individually. These studies usually analyze just the state-of-the-art methods available at publishing, so they were used to isolate promising techniques quickly. Furthermore, the surveys were compared with each other to understand the field's evolution over time. This evolution could be used to identify promising techniques in the field further. For example, if a method was encountered multiple times in multiple papers, this was a strong indication that that specific model had not become obsolete and was still in use, i.e., the method had passed the test of time. These surveys also explain related concepts in automated negotiation, e.g., preference estimation, strategy prediction, and opponent classification [3], which are not analyzed by this study directly but are still relevant. However, these surveys are limited in their comparison capabilities due to the qualitative analysis used in them.

The second type of related work contains studies that analyze different opponent models from a quantitative perspective [4; 19]. They employ numerical evaluation in order to compare the different models directly. Baarslag et al. [4] have analyzed multiple opponent models submitted to the Automated Negotiation Agents Competition [2], and also the metrics used to analyze the accuracy of such a model. The study has identified three categories of opponent models: Bayesian models, Frequency models, and Value models. The researchers have found the best performing model in each category: IAMhaggler Bayesian Model [18], Smith Frequency Model [17] and, CUHK Value Mode [9], respectively. Furthermore, the same study concluded that these "best models are close to being perfectly accurate, which means there is only limited improvement concerning performance" [4]. Note that the best Bayesian model had almost half the accuracy of the other two methods. Their results may indicate that the SOTA algorithmic methods outperform the machine learning techniques. However, the study failed to analyze more advanced machine learning methods, so there is a gap in the available research.

3 Methodology

3.1 Formalizing negotiation

A bilateral negotiation is defined as the exchange of bids between two agents. One agent starts the negotiation session by offering a bid. The opponent can accept it or offer another bid back, i.e., a counter-offer. The negotiation continues until one of the agents accepts the opponent's bid. This can pose a problem, as it is not guaranteed that the negotiation session will end. In order to solve this problem, a time-based deadline is introduced to each negotiation round. In case the deadline is reached, the negotiation ends without consensus. Note that some agents are aware of this deadline and will change their strategy if the deadline approaches.

In order to facilitate easy communication between the negotiating parties, the agents need to agree on the structure of the bid before the start of the negotiation session. A bid ω is defined as a n-dimensional vector of issues $\omega = [\omega_1, \omega_2, \omega_3, ..., \omega_n]$, each issue ω_i having m_i possible values from a given domain : $\forall i \in [1, n] \implies \omega_i \in [\omega_i^1, \omega_i^2, ..., \omega_i^{m_i}]$. The negotiation domain, i.e., bid space: Ω is defined as the set containing all possible bids that an agent can send in a specific negotiating session. The domain Ω is known by both parties before starting the negotiation.

The agents need the capability of comparing different bids with the scope of ranking them, as is the case with any negotiating session. In order to solve this problem, a utility function U is introduced, which evaluates a bid numerically. This utility function maps any bid to a value between zero and one, i.e., $\forall \omega \in \Omega \implies U(\omega) \in [0, 1]$. The utility function can be expressed as a linear combination between a set of weights $[w_1, w_2, ..., w_n]$ and the evaluation of the issue values $[U(\omega_1), U(\omega_2), ..., U(\omega_n)]$, as seen in Equation 1:

$$U(\omega) = U(\omega_1, \omega_2, ..., \omega_n) = \sum_{i=1}^{n} w_i * U(\omega_i) \quad (1)$$

The utility function U represents an agent's preference model. This preference profile is unique for each domain and two different agents usually do not share the same preference profile. Furthermore, the preference profile is kept private by both agents in order to avoid exploitation.

3.2 Opponent modeling

The goal of opponent modeling is to estimate the utility function of the opponent based on the agent's bid history. Note that the bids are received during the negotiation and not before, so the accuracy of the opponent model will gradually increase as the negotiation develops and more bids become available (i.e. the opponent starts revealing more about himself).

The models that will be analyzed and compared in this paper are presented below:

Smith Frequency Model

The Smith Frequency Model [17] will be used as the SOTA algorithmic baseline, as it converges quickly to a high accuracy opponent model. This is enabled by the assumption that the opponent's bids always have maximal utility (i.e. the opponent's utility for the bids his offering is always equal to one). Barslaag et. al. [4] have found that these assumptions can be detrimental in some cases, more specific when the deadline approaches and the opponent starts to change its strategy conceding more making our initial assumption false.

The Smith Frequency Model works by counting how many times the opponent changes each bid but also the frequencies of the issue values. The model then makes the assumption that the issues that change the least are the most important for our opponent, so they will have the biggest weight. Furthermore, the model evaluates the issue values proportionally with their frequency in the opponent's bids.

Perceptron Model

The Perceptron opponent model, proposed by Zafari et al. in 2016 [19], will be used as the machine learning technique to be compared with the algorithmic baseline. This model does not make any assumptions about the opponent's utility value. Furthermore, at each step the model updates just the issue values of the current bid, so the model is resistant to decay, but it will converge slower towards the right value. However, the proposed technique is part of the supervised learning class of machine learning techniques. This means that they require labeled data in order to be trained. This label data consists of the opponent's actual utility function, information that is not available during the negotiation.

The Perceptron model overcomes the need for the opponent's utility by requiring an estimation instead. Estimation of the opponent's utility is a problem solved by opponent modeling, but that is not an option here since the model has not yet been trained. In order to overcome this problem, a rough estimation is needed, which will impact the accuracy of the model.

In order to make our comparison independent of the choice of estimation, we propose two baseline models: the Bad Perceptron and the Perfect Perceptron. The first model will always assume that the opponent's bids have maximal utility, i.e., $U(\omega) = 1$. This assumption is not usually true so this model's accuracy will be used as a lower bound in the comparison. On the other hand, the Perfect Perceptron will have access to the opponent's actual utility, so this model will be used as a theoretical upper bound in the comparison.

Finally, the implementation of the model is slightly changed in order to increase its accuracy. In the original paper [19], the evaluation of the issue values is initialized with a value of 0.5. We found that this initialization produced bad results against agents that lowered their utility function, especially when they start offering bids with utility under 0.5. When this happens, the Perfect Perceptron model starts classifying these seen bids as having less utility than all of the unseen bid space. In order to overcome this problem, the Perceptron model was initialized with a value closer to 0, more specifically 0.1.

3.3 Evaluating the opponent model

Prior comparisons [4] have found there is a linear correspondence between the accuracy of the opponent model and the performance of the agent that incorporates the said model. However, this does not hold for all accuracy metrics, so the proper accuracy measures need to be used.

The opponent model will be tested in isolation using the Pearson correlation of bids [11], which can be seen in Equation 2:

$$d_P(U,U') = \frac{\sum_{\omega\in\Omega}(U(\omega)-\overline{U})(U'(\omega)-\overline{U'})}{\sqrt{\sum_{\omega\in\Omega}(U(\omega)-\overline{U})^2 \sum_{\omega\in\Omega}(U'(\omega)-\overline{U'})^2}} \quad (2)$$

where U denotes the actual utility function of the opponent, U' denotes the predicted utility function (i.e. the utility function computed by our algorithm), $\overline{U}$ denotes the average utility of the opponent over the whole bid space $\overline{U} = \frac{\sum_{\omega\in\Omega} U(\omega)}{|\Omega|}$, and finally, $\overline{U}$ denotes the average estimated utility of the opponent over the whole bid space $\overline{U'} = \frac{\sum_{\omega\in\Omega} U'(\omega)}{|\Omega|}$.

4 Implementation and Experiment

4.1 Environment

A negotiation environment was needed in order to facilitate easy communication between the agents. The environment enforces some of the negotiation constraints and also randomizes the domain and opponents used in each session. The environment was built using the GENIUS framework [12], which has become a standard in the field.

The environment consists of a list of domains, a list of opposing agents, and a deadline. At the start of a negotiating round, a random domain, preference profile, and opponent are picked. This ensures that the experiment is performed in a random manner, on multiple distinct permutations, increasing the generality of the results. After the round is started, the environment is tasked with exchanging the bids between the agents, but also to end the negotiation session in one of two cases: an agent accepts the opponent's bid or the deadline is reached. After the end of the negotiation session, the environment is reset—a new opponent-domain pair is generated and a new preference profile is picked for both agents.

Each experiment will take into account the results over multiple negotiation sessions. This implementation allows for easily setting up a new experiment in which we use just a subset of the available domains and opponents. This allows for a more granular exploration of the behavior of different opponent modeling techniques against different classes of opponents and domains.

4.2 Domain and Preference Profile

In order to further increase the generality of the results, the domain was also randomly generated using the GENIUS framework [12]. Firstly, the different domains vary in the number of issues per bid and also in the number of available values per issue. Note that the issue values are not randomly generated, as these values are not important just the evaluation that the agent applies to these values is important.

Secondly, two preference profiles are generated for each domain. The preference profile consists of: the evaluation of the issue values $U(\omega_i)$ and the issue weights w_i. These values are used by the agents to compute the utility values of the bids, as seen in Equations 1. The preference profiles are also shuffled at the start of each negotiation session, in order to further increase the generality of the experiment.

4.3 Agent

In order to test the opponent model as close to real-life conditions as possible, an automated negotiation agent had to be implemented. The agent needs to be able to receive bids, evaluate them, and decide if it accepts the offer or generates a counter-offer. This main functionality was achieved with the help of the following components, shown in Figure 1.

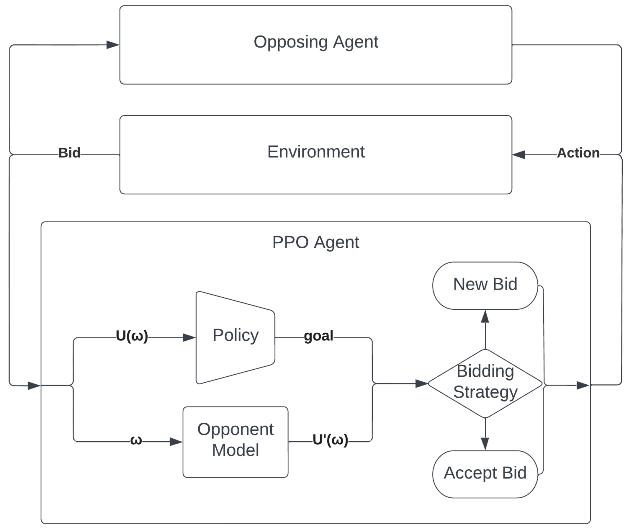

Figure 1: Diagram that shows how the agents interact with the environment and the components of the PPO agent

Opponent Model

The opponent model gradually receives the bids of the opponent and generates an estimated utility function. In total, three classes are implemented, one for each opponent model discussed in the previous section: Smith Frequency model, Bad Perceptron model, and Perfect Perceptron model. All classes are run in parallel, with the scope of comparing their accuracy given the same sequence of bids.

Policy

The policy is a neural network that dictates the behavior of the agent based on the bid history and is implemented using the Proximal Policy Optimization algorithm proposed by Schulman et al. [14]. The policy receives as input the utility of the last four bids and will output the goal for the agent in the form of two values: the agent's utility goal and the opponent's utility goal.

The policy needs to be trained, so the two distinct negotiation domains are created: training and testing. This separation ensures that any high results are due to the agent's capability to be applied to a general negotiation problem and not due to overfitting.

Bidding Strategy

The bidding strategy combines the previous two components in order to decide if the opponent's offer should be accepted or not. Firstly, the bidding strategy calculates for the current bid both the agent's utility and also the estimated utility of the opponent. Secondly, these values are compared with the goals generated by the policy. If the utilities of the current bid are higher than the goals, the current offer is accepted. If the bid does not satisfy this criterion, a random counter offer will be generated and sent to the environment. The counter-offer is picked by randomly selecting bids until one that satisfies the goal is found. Note that it is not guaranteed that such a bid exists, so this process will end after a set number of iterations by returning the closest bid to the goal that was found so far.

4.4 Opposing agents

To simulate a realistic environment, we tested our agent against multiple opponents, each with a different strategy. This ensures that our agents will experience a wide range of bidding strategies, and in turn, our opponent modeling techniques will be tested against a wide range of behaviors. In total, we tested against four agents, all of which use a time-based strategy. This means that all agents will start with a utility goal of one, which is lowered as the negotiation progresses. The utility goal is used by the agents to decide which bids are acceptable or not. When an agent receives a new bid, it will compare its utility with the current goal. If the utility is higher than the goal, the agent will accept the offer. If the utility of the bid is not sufficient, the agent will generate a counter-offer that satisfies this requirement. If more than one bid satisfies this criterion, the agent will pick one at random. This bidding strategy does not take into account any type of opponent modeling, i.e., the opponents don't look at our agent's preferences when picking a bid. The agents lower their utility goal as follows:

- The Hardliner agent will very slowly lower his utility goal.
- The Conceder agent will quickly lower his utility goal.
- The Boulware agent will behave as a Hardliner at the beginning of the round and as a Conceder when the deadline will approach.
- The Linear agent will linearly lower his utility goal.

5 Results

In total, five experiments were conducted, the result of which will be presented below. All the results are calculated over 500 negotiation sessions, using all three models: the Smith Frequency model, the Bad Perceptron model, and the Perfect Perceptron model. Furthermore, each result analyzes just a subset of the available domains and opponents.

The first experiment analyzes the behavior of all models against all four opponents using all domains. The results can be seen in Figure 2, and they show how the accuracy of the different models evolves over the negotiation session. The y-axis indicates the average accuracy of each model, calculated in terms of the Pearson correlation of bids. The x-axis shows the number of bids the opponent has offered, i.e., the number of bids the models have analyzed. Note that these bids are not unique and the opponents can offer the same bid multiple times. The standard deviation of the samples is also shown with a vertical line.

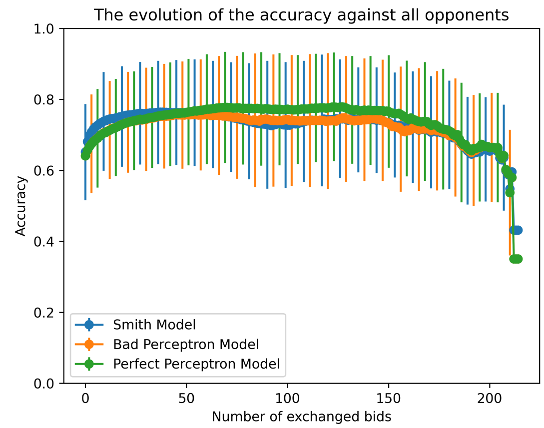

Figure 2: Graph showing the mean and standard deviation of the accuracy, against all opponents.

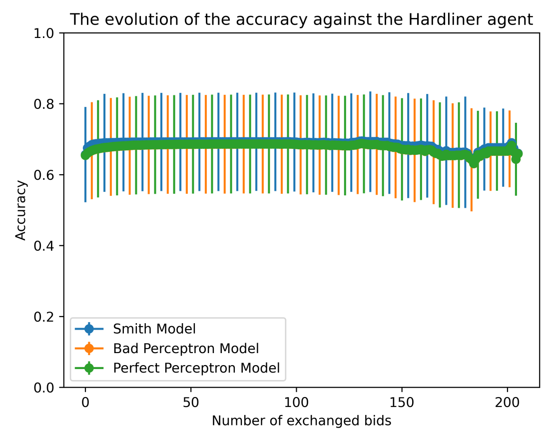

Figure 3: Graph showing the mean and standard deviation of the accuracy, against the Hardliner agent.

The results indicate that the models have similar accuracy. The Smith Frequency has higher accuracy at the beginning of the round, but the other models catch up after 50 bids. Furthermore, the results also show the Perfect Perceptron model to have, on average, higher accuracy than the Smith Frequency mode. However, the difference in accuracy is too small to draw any conclusions, so a more in-depth analysis is required.

The following four experiments will analyze just one type of opponent. The results will be used to identify the preferred model against a specific opponent but also to identify if some opponents are harder to model. Furthermore, these results can also be used to analyze how each opponent's behavior contributed to the results seen in Figure 2.

The first analyzed opponent was the Hardliner agent. The results can be seen in Figure 3. All the models have almost identical behavior against this opponent. This is expected, as the Hardliner always offers bids with maximal utility, so the assumption that both the Smith Frequency and Bad Percep-

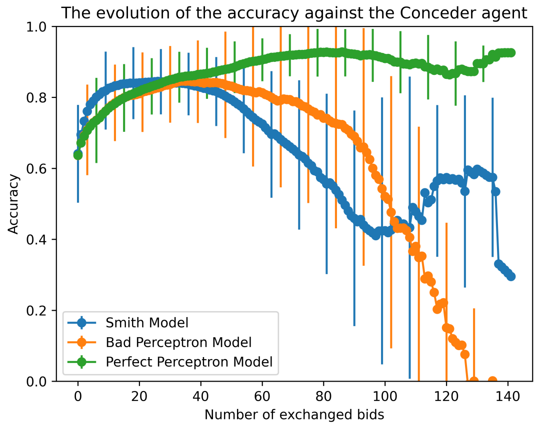

Figure 4: Graph showing the mean and standard deviation of the accuracy, against the Conceder agent.

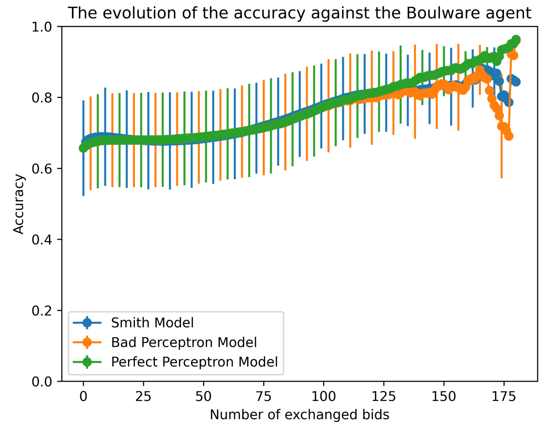

Figure 5: Graph showing the mean and standard deviation of the accuracy, against the Boulware agent.

tron models make is correct. Furthermore, the Perfect Perceptron does not have higher accuracy than the other two models, indicating that the extra information is not advantageous against this opponent.

The second analyzed opponent was the Conceder agent, with the results shown in Figure 4. The assumption that the opponent offers only bids with maximal utility does not hold for this opponent. This does not seem to matter for the first 30 bids, as the models have a similar performance during this period. After the first 30 bids, both models that make this assumption start to degrade, with the Bad Perceptron being more resistant at first. Furthermore, the Bad Perceptron model continues to degrade until it reaches zero accuracy. However, the large standard deviation indicates that the available data is not consistent in the final region of the negotiation. In contrast to the other two models, the Perfect Perceptron can model this type of opponent very well, reaching a high accuracy quickly. This is an indication that the Perceptron model requires an accurate estimation of the opponent's utility for the current bid in order to properly model this opponent.

The third experiment analyzed the Boulware agent. The results for this opponent can be seen in 5. This agent changes its bidding strategy from a Hardliner to a Conceder after approximately 50 bids. We would expect to see the model starting to degrade when this transition happens, as both the Bad Perceptron model and Smith Frequency model had trouble modeling the Conceder agent. However, the results show that this is not the case, as all three models perform very similarly to each other reaching an almost perfect accuracy. It is interesting to note that the Perfect Perceptron does again not benefit in a meaningful way from the extra information it has access to.

The last experiment was conducted on the Linear agent, with the result being shown in Figure 6. The results show that the Smith model dominates at the start of the round, but this domination is short-lived with the Perfect Perceptron Model overtaking Smith after around 50 bids. Furthermore, the Bad Perceptron is very close in terms of accuracy to the Smith model the most of the round, with the notable difference be-

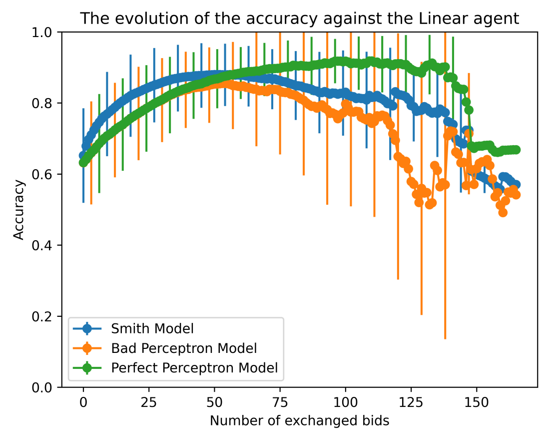

Figure 6: Graph showing the mean and standard deviation of the accuracy, against the Linear agent.

ing located at the end after 100 exchanged bids.

It is interesting to note that all the models performed better against the opponents who lowered their utility goal over time. This may seem counter-intuitive, as these opponents do not follow the assumption that their bids always have maximal utility. We believe the better performance is due to the ability of these agents to offer a larger number of unique bids. As an agent lowers his utility goal, more of the bid space will become available, and in turn, more unique bids can be offered.

In order to test this hypothesis, the evolution of the explored bid space has been plotted in Figure 7 for each opponent. The percentage of the seen bid space can be seen on the y-axis. The x-axis encodes the same information as before, the number of exchanged bids.

However, a strange behavior can be observed, which has been marked with a vertical line in Figure 7. The percentage of the bids space goes down for all opponents at a later stage in the negotiation, which should be impossible. An opponent

Figure 7: Graph showing the time evolution of the average bid space explored by all opponents.

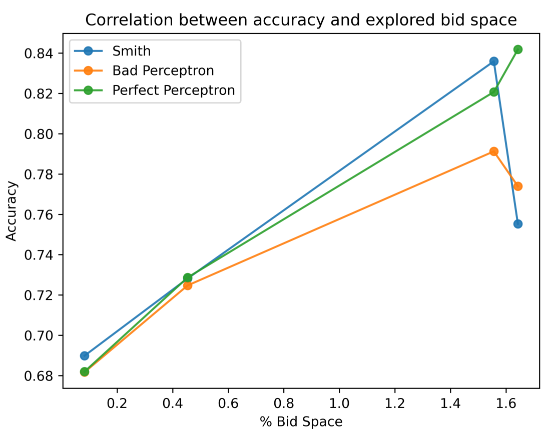

Figure 8: Scatter plot showing the correlation between the accuracy of a model and the explored bid space, for each model.

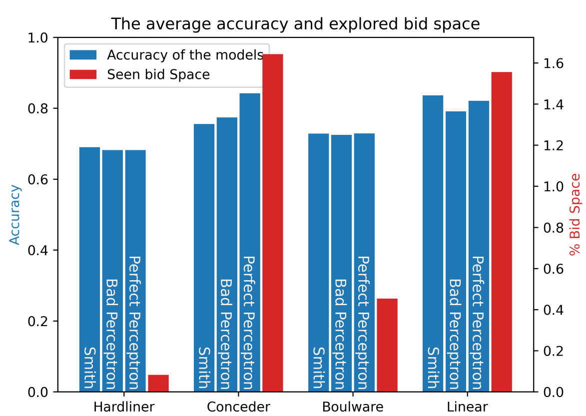

Figure 9: Graph showing the average accuracy against each opponents and the average bid space explored by each opponents.

can either offer a bid that was seen before or a bid that was not seen before, so the percentage of the bid space that was seen can either remain the same or grow. This behavior is due to some negotiations ending earlier by reaching a consensus, so less data is available towards the right side of the graph. Furthermore, the vertical lines may indicate a good ”cutoff” for the date, as they are situated roughly around the time mark at which less than 20% of the initial negotiations are still active.

To further test the hypothesis that a model can better map an opponent that explores more of the available bid space, all the previous results have been averaged in Table 1. The average accuracy of each model has been calculated for all opponents. Furthermore, the average percentage of the explored bid space was also calculated for all opponents. The results seem to indicate that the models have, on average, higher accuracy against the opponents which explore more of the bid space. The results were also plotted in Figure 8. The x-axis shows the average explored bid space. The y-axis shows the average accuracies of the models. It is easy to see from this visualization that the values are highly correlated. So, the results seem to indicate that the opponents that offer more unique bids, and in turn reveal more about themselves, are easier to model.

In order to test the statistical significance of our results, the Pearson correlation coefficient and the p-value have been calculated for the numbers in Table 1. The results for the Smith Frequency model, Bad Perceptron model, and the Perfect Perceptron model are as follows: $(r = 0.79;\ p = 0.20)$, $(r = 0.96;\ p = 0.03)$ and, $(r = 0.99;\ p = 0.005)$, respectively. The results indicate that there is a significant $(p > 0.05)$ large positive correlation $(r > 0.5)$ between the accuracy of both Perceptron models and the percentage of the seen bid space.

In order to further analyze the results shown in Table 1, they are also displayed in Figure 9. Note that the graph has a dual y-axis, with the accuracy being displayed on the left and the explored bid space on the right. The results seem to indicate that even with a significant increase in the percentage of the explored bid space, the corresponding gains in accuracy are minimal.

6 Discussion

Firstly, we will analyze how the Smith Frequency model compares to the Bad Perceptron Model. The results in Figure 2 indicate that both models perform similarly, with the Smith model dominating the start of the negotiation. This is expected, as both models make the same assumption: the opponent's utility is maximal. However, this assumption is used differently by the two models. In the case of the Smith Model, the assumption is directly used in the implementation and can not be easily changed. This is not the case for the Bad Perceptron Model, which does not need this assumption directly, and it is only used to estimate the opponent's utility for the current bid. This assumption is not particularly advantageous for the Perceptron model, as better estimations could be used. It is interesting to note that the two models perform similar, even if the assumption that both models make is to the detriment of the Perceptron model.

	Hardliner	Conceder	Boulware	Linear
Smith Frequency	0.69	0.75	0.73	0.83
Bad Perceptron	0.68	0.77	0.72	0.79
Perfect Perceptron	0.68	0.84	0.73	0.82
Explored Bid Space	0.08%	1.64%	0.45%	1.55%

Table 1: The first three rows show the average accuracy of each model against each opponent. The last row shows the average explored bid space for each opponent

Secondly, we will look at how the Smith Frequency model compares to the Perfect Perceptron model. Most of the results seen in Figure 2 - 6 seem to indicate that a Perfect Perceptron model always outperforms the Smith Frequency model, with a notable exception being the start of the negotiation. This was somewhat expected, as the algorithmic methods are known for their rapid convergence at the start of the round, but is still impressive that the Smith Frequency model manages to outperform another technique that has access to perfect information. However, the Perfect Percetpron always catches up after no more than 30 rounds, overtaking the Smith model in Figure 2 and Figure 4. Furthermore, the results shown in Figure 8 and Figure 9 also indicate that the Perfect Perceptron performs, on average, similar to or better when compared to the Smith Frequency model. The results suggest that the Perceptron model has the potential of outperforming the Smith Frequency model, given that a good enough estimation of the opponent's bid is given.

Next, we will compare the two Perceptron models. The Perfect Perceptron model outperformed the Bad Perceptron model in every experiment, as expected. However, comparing these two models still holds value, as it can help us understand in which cases the additional information helped the Perceptron model to better identify the opponent's preferences. Looking at Figure 9, it seems that the perfect information is only helpful against the Conceder and Linear agent, more so for the first opponent. This is expected, as the assumption made by the Bad Perceptron model does not encapsulate the behavior of these opponents.

Finally, we will discuss how the behavior of different opponents influences the accuracy of the models. We have found a statistically significant positive correlation between the accuracy of the Perceptron model and the percentage of the bid space that was explored by the opponent. This would indicate that the accuracy of this model is highly dependent on the opponent's behavior. However, we have also found that a high increase in the seen bid space corresponds to only a small increase in the model's accuracy. This could be problematic for the further development of this model, as small increases in accuracy would require a massive increase in the explored bid space.

7 Responsible Research

Two distinct problems were taken into account when considering the responsibility of this research. On one hand, the reproducibility of the research is always a concern, so extra care has been taken to assure the reproducibility of this research. On the other hand, the ethical aspects of the opponent model have also been explored.

One of the most encountered problems by scientists, in terms of reproducibility, is not having access to the original code-base of the paper. Having access to the original code-base can prove very useful, as it allows researchers to quickly reproduce the results, but also to identify possible flaws in the original setup. To mitigate this problem, the code used to generate the results was made publicly available [1].

One of the last identified problems, in terms of reproducibility, was the use of randomizing functions. Random functions are usually used to increase the generality of the problem, by shuffling the testing domain. This can pose a problem, as two consecutive running of the code-base might produce different results. To solve this problem, a seed was introduced to the random number generator. This ensures the reproducibility of the results in consecutive runs but also enables the benefits of using a random function. Furthermore, the seed was left unchanged in the final repository.

The ethical aspects that were analyzed are related to the opponent's exploitation. Opponent modeling is required because the agent like to keep their preference private in order to avoid exploitation. This can pose a problem, as the current procedure can model the opponent quite well. If the current models keep improving, the agent might become vulnerable to the same exploitation we are trying to avoid. In order to mitigate this, the agents could study their opponent's bid history, in order to identify any trends. A negative trend could indicate that the opponent is using the information gathered during the negotiation maliciously, i.e., the opponent is exploiting us.

8 Conclusions and Future Work

This paper analyzes how machine learning compares to the SOTA heuristic techniques when used to model the opponent's preferences. Historically, heuristic techniques have dominated the field, but we have found this to no longer be valid. Our main conclusion is that machine learning techniques are at least as good as their algorithmic counterparts when estimating the opponent's preferences in a negotiation problem.

Furthermore, we believe that further research has the potential of increasing the accuracy of these techniques even more. Previous research has concluded that heuristic models have just limited room for improvement. However, we have shown that machine learning techniques can outperform these heuristic models, given the right circumstances. Therefore, we believe that machine learning techniques still have

[1]https://github.com/brenting/negotiation_PPO/tree/opponent-models-comparison

room for improvement, with the potential of increasing the accuracy of the opponent model even higher than the current state-of-the-art.

Finally, we have also found a statistically significant correlation between the accuracy of the machine learning models and the percentage of the bid space that was explored by the opponent. This can pose a problem, as this would suggest that the model's accuracy is limited by the opponent's behavior. Based on this, we believe that these models are approaching their theoretical limit, as they do not have a way of influencing the explored bid space.

References

[1] Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, May 2018.

[2] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198:73–103, May 2013.

[3] Tim Baarslag, Mark J. C. Hendrikx, Koen V. Hindriks, and Catholijn M. Jonker. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30(5):849–898, September 2016.

[4] Tim Baarslag, Koen Hindriks, Mark Hendrikx, and Catholijn Jonker. Predicting the Performance of Opponent Models in Automated Negotiation. *Proceedings - 2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2013*, 2, November 2013.

[5] Carrie Beam and A. Segev. Automated Negotiations: A Survey of the State of the Art. *Wirtschaftsinformatik*, 30(3):263–268, 1997.

[6] Christie Etukudor, Benoit Couraud, Valentin Robu, Wolf-Gerrit Früh, David Flynn, and Chinonso Okereke. Automated Negotiation for Peer-to-Peer Electricity Trading in Local Energy Markets. *Energies*, 13(4):920, February 2020.

[7] Jakob Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with Opponent-Learning Awareness. In *AAMAS '18: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130. International Foundation for Autonomous Agents and Multiagent Systems, July 2018.

[8] Andreas Riener Franz Keferböck. Strategies for Negotiation between Autonomous Vehicles and Pedestrians. In *Mensch und Computer 2015 – Workshopband*, pages 525–532. De Gruyter, Berlin, Germany, September 2015.

[9] Jianye Hao and Ho-fung Leung. CUHKAgent: An Adaptive Negotiation Strategy for Bilateral Negotiations over Multiple Items. In *Novel Insights in Agent-based Complex Automated Negotiation*, pages 171–179. Springer, Tokyo, Japan, January 2014.

[10] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé. Opponent modeling in deep reinforcement learning. In *ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 1804–1813. JMLR.org, June 2016.

[11] Koen V. Hindriks and Dmytro Tykhonov. Towards a Quality Assessment Method for Learning Preference Profiles in Negotiation. In *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, pages 46–59. Springer, Berlin, Germany, 2010.

[12] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M. Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1):48–70, 2014.

[13] Samer Nashed and Shlomo Zilberstein. A Survey of Opponent Modeling in Adversarial Domains. *Journal of Artificial Intelligence Research*, 73:277–327, January 2022.

[14] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *ArXiv e-prints*, July 2017.

[15] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018.

[16] Leigh Thompson. *The Mind and Heart of the Negotiator*. Pearson Education, Toronto, Ontario, Canada, 2012.

[17] Niels van Galen Last. Agent Smith: Opponent Model Estimation in Bilateral Multi-issue Negotiation. In *New Trends in Agent-Based Complex Automated Negotiations*, pages 167–174. Springer, Berlin, Germany, November 2011.

[18] Colin R. Williams, Valentin Robu, Enrico H. Gerding, and Nicholas R. Jennings. IAMhaggler: A Negotiation Agent for Complex Environments. In *New Trends in Agent-Based Complex Automated Negotiations*, pages 151–158. Springer, Berlin, Germany, November 2011.

[19] Farhad Zafari and Faria Nassiri-Mofakham. POPPONENT: Highly accurate, individually and socially efficient opponent preference model in bilateral multi issue negotiations. *Artificial Intelligence*, 237:59–91, August 2016.