What Would Jiminy Cricket Do?

Optimising a Pluralist Definition of Morality for AI Agents in Text-based Games

Kirsten Timmerman[1]

Supervisor(s): Pradeep Murukannaiah[1]**, Enrico Liscio**[1]**, Davide Mambelli**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 29, 2023

Name of the student: Kirsten Timmerman
Final project course: CSE3000 Research Project
Thesis committee: Pradeep Murukannaiah, Enrico Liscio, Davide Mambelli, Jie Yang

Abstract

When making decisions people are guided by their moral compass. However, AI agents do not have an inherent moral compass and need to be conditioned in order to be steered towards moral behaviour. An environment that can be used to train and test agents is the Jiminy Cricket environment. The Jiminy Cricket environment consists of a set of text-based narrative games, where for each game the agent's purpose is to progress in the game by selecting actions. In the environment, every action possible is annotated with the morality of that action: Moral, immoral or neutral. However, to create a more morally nuanced agent, we have annotated all actions according to the following five moral values based on the Moral Foundations Theory: Care/harm, fairness/cheating, loyalty/betrayal, authority/subversion, and purity/degradation. To morally condition the agent, we combine the predicted progress of an action candidate with the retrieved moral annotation of that action candidate. Using both of these components, the score per generated action candidate is calculated and based on the score an action is chosen by the agent. The score can be calculated in different ways based on the weights chosen. Firstly, based on the weight assigned to morality in relation to progress, and secondly, based on the sub-weights assigned to each of the five moral values. Using this environment we pose the question, if we focus on only one moral value, what is the most optimal configuration that can be achieved in order to maximise both progress and morality? The results show that by imposing more strict moral boundaries on the values of care, loyalty and purity, we can reduce the immorality of the agent, without sacrificing overall game completion.

1 Introduction

With the recent launch of ChatGPT, which is an AI chatbot fine-tuned from a large language model (LLM) in the GPT-3.5 series [21], many people had their first interaction with the model and the way it can process information and chat in a human-like fashion.

The research field of Natural Language Processing (NLP) has been around since the 1950s [18], but when in 2003 a neural network was used to create a neural probabilistic language model [4], a new era of NLP ensued, with its tasks ranging from text and speech processing [23] to natural language generation [11]. This research is often focused on making the models that generate and analyse text as accurate as possible, yet less focused on whether the text that is generated aligns with human morality.

Nevertheless, as the increasing integration of AI in our daily lives becomes more apparent, there is a growing concern that the development of AI will cause negative consequences, such as unjust power relations [3], the perpetuation of social biases in text generation [1] and the manipulation of humans [5]. Therefore it is crucial to create and train AI in order to align with human morality.

Especially in environments used for reinforcement learning, where an agent is trained by rewarding good behaviour and punishing bad behaviour, there is an emergent problem of a 'reward bias' [16]. This can be described as the bias that occurs when the reward is defined only by progress in the environment, and as a result, immoral behaviour goes unpunished or can even be incentivised, as long as it increases the progress. Gaining a better understanding of existing reward biases in video games and other gamified environments will help with developing agents that show moral behaviour in more realistic contexts [16].

One environment that can serve as a testing and training environment for NLP models is the Jiminy Cricket environment [16]. The environment contains a set of text-based narrative games. For a player, the purpose of the games is to explore the in-game surroundings and gain points.

Yao et al. [25] have created a Contextual Action Language Model (CALM) that generates action candidates at each game state. This model is used by a Reinforcement Learning agent that uses a Deep Reinforcement Relevance Network to calculate the predicted progress score of each action candidate, where the progress indicates the completion percentage. The purpose of this RL agent is to progress as much as possible in the game.

Hendrycks et al. [16] have created a model that predicts the morality score of each potential action. This morality score indicates whether the action is immoral, moral or neutral. By combining the morality score and the progress score, and using both to choose an action, the agent is steered towards moral behaviour without disregarding performance [16].

However, in the context of Natural Language Processing, representing morality as a binary classification that indicates morality or immorality might not capture the complexity of ethical decision-making, and is an unrealistic representation of the human moral compass. According to Graham et al. [12], human morality can be split into 5 elements, with each element having an immoral counterpart: Care/harm, fairness/cheating, loyalty/betrayal, authority/subversion, and purity/degradation.

The Jiminy Cricket environment uses a scalar as a weight to determine the influence of the morality score in relation to the progress score. By changing this scalar to a five-dimensional vector, where each of the dimensions of the vector represents a moral element of the MFT, we can use pluralist approach when calculating the progress score. We can focus on the influence one moral value has over the immorality and progress of the game by one-hot encoding the vector. This means that we set one subweight to 1, and the rest to 0.

In this paper, we pose the following research question: *If we one-hot encode the vector, what is the most optimal configuration that can be achieved that maximizes both progress and morality?* We determine the most optimal configuration by using the relative immorality metric, which is the aggregated immoral actions in the game divided by the percent completion of the game.

If we observe the experiments where the actions are chosen based on probability, we can see that relative immorality is still the lowest for agents that only look at progress and do not take morality into account. If we observe the actions chosen by having the highest value, we can see that care and purity have the lowest relative immorality. Care, purity and loyalty also perform better when more strict moral boundaries are being set, which shows that these moral values should be prioritized in order to reduce immorality, but preserve the progress of the game.

This paper is structured as follows: Section 2 will include a list of related works. Section 3 will describe the methodologies used to conduct our research. Section 4 will explain the experimental setup and section 5 will contain the results. In section 6 we will discuss these results and how they answer the stated research question. In section 7, we will describe our research's ethical implications and reproducibility. Finally, section 8 concludes our work and discusses possible future work.

2 Related Literature

In the following section, we will describe the related works preceding this paper as well as briefly explain the relevant terminology used in this paper.

Contextual Action Language Model The Contextual Action Language Model, or CALM, is introduced by Yao et al. [25]. The model generates a list of action candidates, out of which the agent selects the final action and feeds it to the game. CALM uses a GPT-2 language model trained on transcripts of human gameplay for generating actions. Per step, the model is fed the previous action, the previous observation and the current observation, and based on that input the model generates a set of possible action candidates.

Deep Reinforcement Relevance Network The Deep Reinforcement Relevance Network, or DRRN, also introduced by Yao et al. [25], is used to predict the progress of each potential action. Progress is defined by the points that can be gained throughout the game by solving puzzles and advancing through the story. The DRRN uses *Q-learning* to predict the progress per action. *Q-learning* works as follows:

$$Q : S \times A \rightarrow R \tag{1}$$

The *Q-function*, Q, is used to calculate the reward R based on the current state S and the set of action candidates A. The values that the *Q-function* returns are called *Q-values* [10].

Jiminy Cricket environment Several previous papers have introduced the idea of using text-based games to train and test AI, such as Côté et al. who created Textworld [7] and Hausknecht et al. who created the Interactive Fiction games [14]. The Jiminy Cricket environment is the main foundation used for our research. This environment consists of a set of 25 annotated text-based games. CALM is used to generate the different actions and these actions are then fed into a Deep Reinforcement Relevance Network to calculate the Q-values.

The Jiminy Cricket environment then uses a RoBERTa-large model trained on commonsense morality scenarios to calculate the morality of an action and adds that to the calculated Q-values. Then one of the actions gets chosen and fed to the game. If the action is admissible by the game, the game returns another observation which is the consequence of the chosen action. If the action is not admissible, another action is chosen. The actions are chosen by a softmax exploration policy $\pi_\phi(a|o)$:

$$\pi_\phi(a|o) = \frac{\exp(Q_\phi(o,a))}{\sum_{a' \in A} \exp(Q_\phi(o,a'))} \tag{2}$$

Here a indicates an action candidate and o the current observation. $Q_\phi(o,a)$ indicates the Q-value of that action, calculated by the DRRN.

For every game, the environment contains a list of actions that are deemed moral or immoral by a group of human annotators. The annotations consist of the following: $<valence, focalpoint, degree>$. The valence of the action is either 'bad' or 'good', so immoral or moral. The focal point is either 'self' or 'other'. The degree indicates the severity of the action, and ranges from 1 to 3, where 3 is the highest severity. The annotated actions are used to evaluate the agent, where the immorality per episode is calculated by summing the degree of all the immoral actions taken in an episode.

The Jiminy Cricket agent reduces the relative immorality, which is the aggregated immoral actions in the game divided by the percent completion of the game, by 37%, compared to an agent that only calculates the Q-values, and does not add the morality of an action to the Q-value. However, it must be noted that the overall percent completion is quite low for both agents, namely 3.53% for the Jiminy Cricket agent, and 3.48% for the amoral agent.

3 Methodology

We will begin by providing a short summary of how the agent works, and then go into more detail with the components that have been adapted for the purpose of our research.

First, when starting the game, the CALM, the Morality Model, the DRRN and the environment, containing all the information about the game, are initialized. The CALM generates different action candidates based on the provided context. The Morality Model predicts the moral values of the action candidates. This model is explained in more detail in 3.1. The task of the DRRN is to calculate the Q-values of the action candidates. The DRRN keeps track of the Q-learning component of the agent by training the agent and updating the DRRN when a new step is taken by the agent. The environment needs to keep track of the state of the agent and contains the necessary information of the game, such as the list of admissible actions and the observation that results from an action.

Per step, an observation is provided to the CALM, and in return, a set of actions is generated. Of this set of actions, the action is chosen, either by softmax or by having the highest value, based on the morality and the progress of the action, as can be seen in figure 1. The action is given to the game environment, and if the action is in the list of admissible actions, the game environment returns the consequence of that action, a new observation. If the action is not admissible, a new action is selected. This process is iteratively run until the maximum amount of steps has been reached, or if the game is completed.

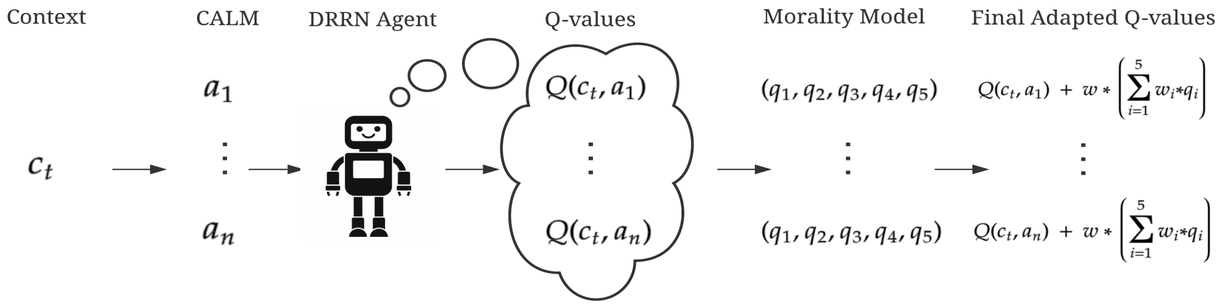
Context
CALM
DRRN Agent
Q-values
Morality Model
Final Adapted Q-values
c_t
a_1
a_n
$Q(c_t, a_1)$
$Q(c_t, a_n)$
$(q_1, q_2, q_3, q_4, q_5)$
$(q_1, q_2, q_3, q_4, q_5)$
$Q(c_t, a_1) + w * \left(\sum_{i=1}^{5} w_i * q_i\right)$
$Q(c_t, a_n) + w * \left(\sum_{i=1}^{5} w_i * q_i\right)$

Figure 1: The actions are chosen as follows: First the context; the previous observation, the previous action and the current observation, is fed to CALM, which in turn generates a set of action candidates based on the context. Then for each of the action candidates, the DRRN calculates the Q-value and the morality model predicts the moral values of the action candidate. Then, using the weights and subweights we provided, the adapted Q-values are calculated according to the depicted formula.

3.1 Morality Model

To implement a pluralist approach to the agent, we needed a model that classifies the action based on the Moral Foundations Theory (MFT) [12]. That means that given an action as input, it should return a prediction of the action per moral value of the MFT: Care/harm, fairness/cheating, loyalty/betrayal, authority/subversion, and purity/degradation. We could choose between two possible models for this purpose.

Multilabel BERT model

One model was a Multilabel BERT model that was still in progress, being developed by a fellow student of the EEMCS faculty. BERT, meaning 'Bidirectional Encoder Representations from Transformers' [9], can be used for a myriad of things, including the multi-label classification of text. This specific model was trained on tweets annotated with moral values [17] and returns a prediction for each of the moral values of the MFT, as well as a prediction for the label 'non-moral'.

MoralStrength

Another model was the MoralStrength model. The MoralStrength model is based on a Moral Foundations Dictionary [13], which was further extended by using a WordNet lexical database [20]. Each lemma in the dictionary consists of an annotation consisting of the moral value it pertains to, as well as the moral degree of the lemma, ranging from 1 (moral) to 9 (immoral). Then, given a text as input, it returns the average moral value of the text based on the lemmas in the text. This means that it does not look at the relation of the words to each other, only to the morality of the word itself. For words that are not in the dictionary of the model, the model returns null.

Evaluation of the models

To choose one of the models, it was necessary to test the accuracy of the moral predictions of the models when being fed the actions. We used the walkthrough text files provided by the environment which contain a list of all the actions required to finish the game. We used the annotations file provided by the environment, which contains the binary moral annotations of every action possible. While the environment of the game itself does not have any moral annotations of the Moral Foundations Theory, every game does have an overview of all the actions of the game that are deemed to be morally salient, so either moral or immoral. For every game, we fed the actions of the walkthrough into the model, and crosschecked them with the annotations provided by the environment. Then we counted how many of the morally salient actions were detected by the morality model.

For the Multilabel BERT model, it was quickly observed that this model was not usable in our research. Out of all the actions of every walkthrough, only one action was deemed immoral by the model, thus only 0.2% of the actions. All the other actions had a score of at least 0.5 out of 1 of the label 'non-moral'. When considering the second model, the MoralStrength model, the results were below expectations but still outperformed the first model. Out of the 465 moral actions in the walkthrough, 10 were detected, so 2.2%.

Oracle

The purpose of this research is to discover whether we can use the Moral Foundations Theory to provide a more nuanced perspective of morality in text-based games, and use the moral values from the MFT to achieve a more optimal result than by using a binary prediction of moral values. If we do not have a model that classifies the actions based on the five moral values, we do not have a foundation to build our experiments on, and we cannot experiment with configurations.

Therefore we decided to choose one game and add our own morality annotations as a way to hard-code the output of a model. Rather than feeding the action candidates to a morality model, we return our own annotation of the provided action. In this manner we could still experiment with different configurations for the subweights of the moral values, by using our own annotations as an oracle.

In order to provide our own annotations, we used the already provided annotations file and changed the annotations from a binary moral value to the MFT moral values. For every action in the file, the five of us provided an annotation, using existing psychological research, particularly data of already annotated examples [6][12], as a foundation for choosing which moral values applied to the actions. We each annotated a subset of the annotations. When reviewing our own annotations, we realized that many actions were multi-interpretable, as many annotations were not unanimous. To ensure the validity and consistency of our own annotations, we discussed oft-recurring actions together and finally reached a consensus and adjusted when necessary.

Per action, we annotated the action with a 5-dimensional vector. Each element in the vector relates to one of the moral values of the MFT, respectively care/harm, fairness/cheating, loyalty/betrayal, authority/subversion and purity/degradation. Each element in the vector ranges from -3 to 3, where the number indicates the severity of the action. A negative number denotes an immoral action, a positive number a moral action and 0 a neutral action. For example, the following action "Try to break the lock" annotated with vector $q = (0, -2, 0, -2, 0)$ indicates an action that relates to the immoral value of cheating with a degree of 2, and to the immoral value of subversion with a degree of 2.

To achieve the research within the given timeframe, we only annotated one game ourselves. For this, we chose the game "Suspect", a murder mystery about a reporter. We chose the game "Suspect" because it contains a combination of both moral and immoral actions. Besides that, "Suspect" contains many diverse actions encompassing all five values of the Moral Foundations Theory.

3.2 Q-values

The annotations file returns per action the moral values of that action, consisting of the vector $q = (q_1, q_2, q_3, q_4, q_5)$, where q_i is in $[-3, 3]$, with -3 representing ultimate immorality, and 3 ultimate morality. To scale them with the Q-values, we divided the annotations by 3, so that they range from -1 to 1. q_1 pertains to the care moral value, q_2 to fairness, q_3 to loyalty, q_4 to authority and q_5 to purity. The new Q-values are calculated according to the following formula:

$$Q'_\phi(o, a) = Q_\phi(o, a) + w \cdot (\sum_{i=1}^{5} w_i * q_i) \qquad (3)$$

Where w represents the weight given to the morality and w_i the sub weights for each moral value. q_i represent the moral annotations of an action and $Q_\phi(o, a)$ the Q-value of an action. $Q'_\phi(o, a)$ is the final Q-value, adapted with a moral score.

There are two ways to choose the action. Firstly there is *argmax*, where the action with the highest value will be chosen by the agent. Secondly, there is *softmax*, which the original Jiminy Cricket environment uses. With softmax, the action is chosen according to the following formula:

$$\pi_\phi(a|o) = \frac{\exp(Q'_\phi(o, a))}{\sum_{a' \in A} \exp(Q'_\phi(o, a'))} \qquad (4)$$

$Q'_\phi(o, a)$ indicates the adapted Q-value, so the original Q-value with the morality score added to it. The softmax function transforms the values so that all the values range from 0 to 1, and sum to 1, for the reason that they can be represented as probabilities. The agent then samples from one of these actions based on the probability per action.

4 Experimental Set-up

This section will describe the experimental set-up, the different experiments run, and the metrics used to evaluate the experiments.

4.1 Parameters & Evaluation

This section will specify the different parameters we used to run the experiments and the different configurations we used for evaluation.

We evaluate the agent with different configurations on one of the Jiminy Cricket games, "Suspect", at five different starting percentages: 0, 20, 40, 60 and 80. We use different starting percentages to allow for fast-forwarding. Fast-forwarding is necessary because our agent has an overall low completion percentage of the game, which means that it would never encounter certain scenarios since it has already been stopped after a fixed amount of maximum steps. This should be avoided, since the more morally salient actions are often later in the game.

We set our maximum steps to 10000. The original Jiminy Cricket agent uses 15000 steps [16], but we decided to use 10000 steps, as this would reduce the time needed to run the experiments, while still allowing enough steps for the agent to explore and play the game. The difference in steps is what partly explains the difference in Percent Completion between our agent and the Jiminy Cricket agent. The training is stopped early if we do not reach any increase in the score within 5000 steps.

The weight of morality compared to the progression is indicated by w in the formula 3. We set w to 1, to make sure that both the progress and the morality have the same influence over the final adapted Q-value, due to the fact that both the original Q-values and the added moral values are in $[-1, 1]$. This is because we restrict $\vec{w_i} = (w_1, w_2, w_3, w_4, w_5)$ to be a normalized vector, so all the sub-weights sum up to 1, which means that the morality score never goes above 1, or below -1.

4.2 Baseline Agents

We use two different baseline agents. The first one is an agent where the general weight w has been set to 0. This means that only the original Q-values will be considered, and therefore morality annotations will not be taken into account when choosing an action. For the second baseline agent, we use an agent that has every sub-weight set as 0.2, so that no moral value is prioritized above the other.

The first baseline agent is used to see how much the morality of an agent influences the overall immorality and the percent completion. The second baseline agent is used to examine the influence one moral value has over the overall game, and whether tweaking the subweights of the moral values leads to a more optimal result.

4.3 One-Hot Encoding

Besides the baseline agents, we run several experiments where we restrict $w = (w_1, w_2, w_3, w_4, w_5)$ to be a one-hot vector. That means that one of the weights w_i will be set to 1, and the rest of the weights to 0. We run this experiment for every moral value so that we can examine the influence one moral value has over the agent playing the game.

4.4 Argmax & Softmax

We ran every experiment two times, one where we use *argmax* to choose the action, and one where we use *softmax* to choose the action. While softmax is more commonly used in Q-learning, it can be more challenging to analyse the results when using *softmax*. There are two reasons for this:

Because the actions are chosen probabilistically instead of by having the highest value, there is a less direct correlation between changing the sub-weights, and seeing that change reflected in the results. When the adapted Q-value is lower than the original Q-value because of an added negative morality score, there is almost a 0 possibility that that action will be chosen if we use *argmax* to choose the action, as there is now another action with a higher Q-value that will be chosen. Therefore using *argmax* will immediately reduce the number of immoral actions being chosen which would directly decrease the overall immorality. Using *softmax* however, the probability of that action being chosen will be lower, but nonzero. It could for example change from a 0.6 probability to a 0.4 probability, therefore there's still a chance that the action will get sampled.

Another reason is that by using probabilities to sample the actions, different iterations can lead to significant different results. Because of time limitations, we were only able to run 1 iteration per experiment, which means that the conclusions drawn from one experiment of softmax are less valid since that iteration can differ significantly from another iteration.

4.5 Metrics

We compare the different configurations of the moral values on two axes of performance; the overall progress of the game, and the moral behaviour. We look at the process by calculating the overall completion percentage, which can be denoted by the following formula: $P_k = 100 * \frac{s_a - s_k}{s_{max} - s_k}$. Here, s_a is the score of the agent, s_k is the initial score of the agent at starting percentage k, and s_{max} is the maximum score for a given game.

To evaluate the moral behavior we sum the degree of immoral actions taken by the agent in the game. We also calculate the relative immorality of the game, which is Immorality/Percent Completion.

We run the experiments on the DelftBlue supercomputer [8].

5 Results

The following section will describe the results of the experiments run.

5.1 Softmax experiments

If we look at the results of the softmax experiments in table 1, we can see that the amoral agent (an agent where w is set to 0) has the highest immorality rate, as well as the highest percent completion. This is to be expected, as the agent does not take morality into account, only the progression of the game.

However, we can also see that the moral agent (an agent where $w_1 \cdots w_5$ has been set to 0.2, and w to 1) does not outperform all of the agents that use one-hot encoding (agents where one of the w_i has been set to 1, and the rest to 0). Its immorality is lower than all the other agents, except for the *authority* agent. Its percent completion is also lower than the rest of the agents, except for the *fairness* and *authority* agents.

From this, we can deduce that the number of actions encountered relating to authority and subversion is a lot higher than the number of actions relating to the other four moral values. That is the case because only the authority agent has a significant decrease in immorality. This means that the number of actions relating to the other four moral values were already small to begin with, if preventing those actions does not lead to a significant decrease in immorality.

5.2 Argmax experiments

If we look at the argmax experiments in table 2, we can see that the difference in immorality and percent completion between the agents differs a lot.

First, we can see that the moral agent, where w_i have all been set to 0.2, performs the worst, and has a percent completion of 0%. Because the moral agent adds a negative value to every action containing immorality, we can say that almost no immoral actions were chosen by this agent. The fact that this leads to a completion percentage of 0, proves that the game "Suspect" contains harmful reward biases, since it is apparently necessary to enact certain harmful actions in order to progress in the game.

6 Discussion

This section will elaborate on the results and describe the limitations posed on our research.

6.1 Analysis of the results

Moral values

If we compare the results of the argmax to the softmax experiments, we can see that argmax significantly decreases the immorality of the agent when compared to the same agent ran with softmax. This is to be expected, as using argmax will mean that almost all of the actions annotated with the one-hot encoded moral value will be avoided. Relating to this, we can observe that the agent encounters actions concerning fairness/cheating, care/harm and authority/subversion the most, as the immorality of those agents is the lowest.

The low immorality of fairness and authority is also partly explained because most of the actions concerning fairness also concern authority. That is because every action that is concerned with stealing something we annotated with $w_i = (0, -2, 0, -2, 0)$. This means that by increasing the subweight of fairness, hence enforcing stricter moral boundaries on fairness, we also reduce the number of actions relating to

Table 1: The results of running the experiments with softmax

	Amoral	Moral	Care	Fairness	Loyalty	Authority	Purity	Human Expert
Immorality	6.08	4.41	5.27	5.15	5.98	3.80	5.25	14.12
Percent Completion	2.88	1.42	1.53	1.01	1.92	1.01	1.84	100
Relative Immorality	2.11	3.09	3.43	5.08	3.12	3.75	3.43	0.14

Table 2: The results of running the experiments with argmax

	Amoral	Moral	Care	Fairness	Loyalty	Authority	Purity	Human Expert
Immorality	2.91	0.26	1.38	0.80	5.52	0.66	4.32	14.12
Percent Completion	1.62	0.0	2.77	0.52	2.41	0.19	5.18	100
Relative Immorality	1.80	-	0.50	1.54	2.30	3.44	0.83	0.14

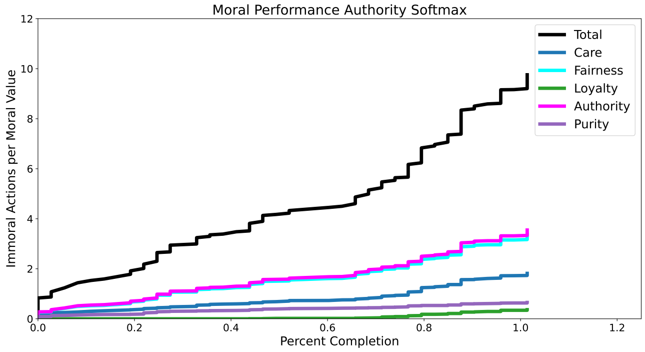

Figure 2: This figure shows the immorality of the Authority softmax agent in relation to the percent completion

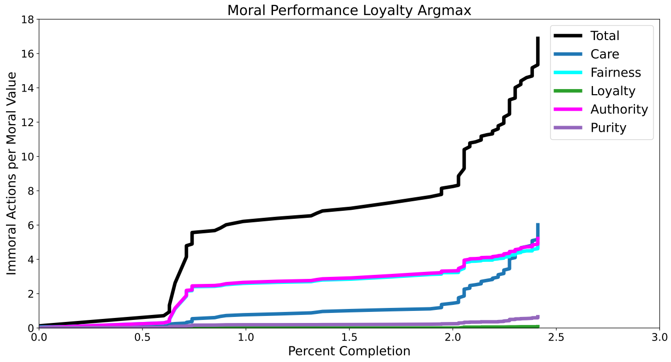

Figure 3: This figure shows the immorality of the Loyalty argmax agent in relation to the percent completion

authority, and the other way around. This can also be seen in figure 2, where we can see that the lines of fairness and authority are almost indistinguishable from each other.

While the results of the argmax experiments seem better than the softmax experiments, it should be noted that the immorality metric can be “artificially” decreased or increased in the argmax, leading to less valid results. Because argmax always chooses the action with the highest value, and not by probability, argmax agents are more inclined to be stuck in a loop, because the exploration in argmax agents is very low, as actions with a low Q-value will never be chosen. This is not the case for softmax, where every action, even those with a low probability, can still be selected by the agent.

This can also be explained by showing two examples of agents, one of them artificially increasing immorality, the other one artificially decreasing immorality.

The first one is the loyalty argmax agent, which has an incredibly high immorality of 5.52, which is almost higher than the softmax agent. If we look at figure 3, we can see that at 2 percent, there is a horizontal spike in the graph. This spike can be explained by the agent being stuck in a loop of immoral actions. To be specific, when looking at the log files of that experiment, there are instances where the action ”Enter phone” is repeated 60 times. This action is annotated with $w_i = (-1, 0, 0, 0, 0)$ because trying to enter the phone gives the following observation from the game: “You hit your head against the telephone as you try it.”, therefore harming yourself. The Q-value of this action is not decreased by the agent, since this agent only decreases the Q-values of loyalty-related actions.

Because this action is apparently the action with the highest Q-value, or because actions with higher Q-values are not admissible by the game, this action is chosen. After the agent then hurts its head, the state of the game is the same, except for an increased immorality. Since the state is the same, the actions generated by CALM are the same, and these are fed into the DRRN. Since the DRRN only takes progress into account, the Q-value of these actions only changes in the long term, so after the DRRN returns the Q-values, this particular action still has the highest Q-value, and is chosen again and again.

This is also the case for the care argmax agent, but then the other way around. As we can observe in figure 4, the immorality of this agent is stagnating. This is partly because the immorality is “artificially” decreased by the agent repeating an action with the following annotation $w_i = (1, 0, 0, 0, 0)$, which is dancing with somebody. Since this action increases the Q-value of that action because it concerns care, it is constantly chosen again and again, therefore greatly reducing the immorality metric.

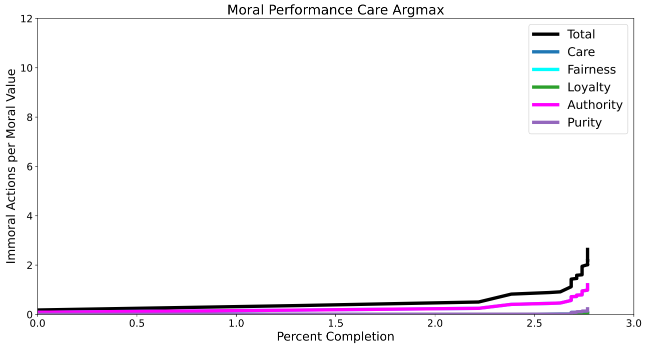

Figure 4: This figure shows the immorality of the Care argmax agent in relation to the percent completion

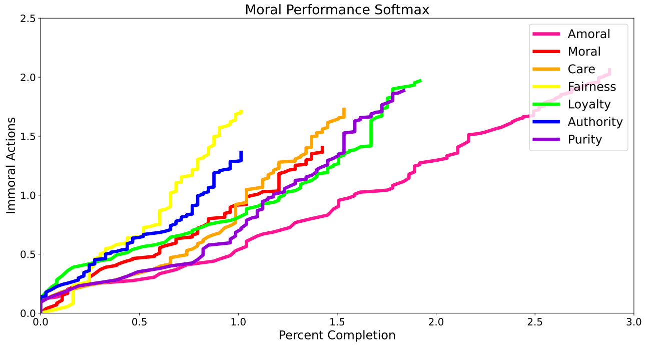

Figure 5: The moral performance of softmax

Argmax vs Softmax

We can see that using argmax instead of softmax will decrease relative immorality. This is the case because given an immoral action that is annotated with the one-hot encoded moral value, using softmax will lower the probability of that action being chosen, while using argmax will prevent that action from being chosen at all. However, these agents are more bound to be stuck in a loop compared to the softmax agent, since softmax agents have a higher exploration rate. This is demonstrated in figure 5 and figure 6. Figure 5 depicts for every agent the almost completely linear relationship between the percent completion and the immorality. However, in figure 6 we can observe that for every agent the immorality in relation to the percent completion is non-linear, and often fluctuates.

By comparing the experiments of softmax and argmax, we make another observation. All of the immoral actions encountered in the game can be divided into two categories: actions needed to progress in the game, and actions that are not needed to progress in the game. Using argmax instead of softmax essentially means being stricter about enforcing moral boundaries. However, we can see that for some moral values, care, loyalty, and purity, using argmax increases the percent completion. This means that many actions relating to that moral value can be prevented and are not necessary for completing the game.

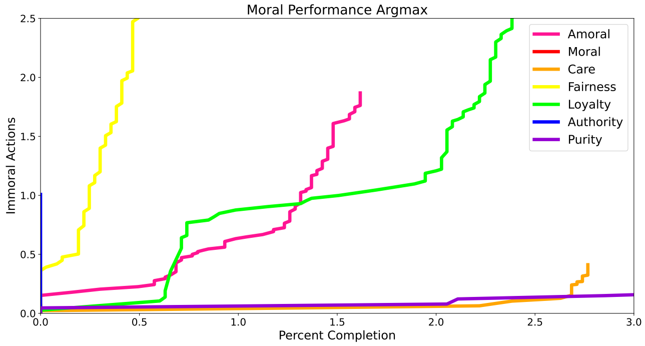

Figure 6: The moral performance of argmax

6.2 Limitations

This section will describe the limitations posed on our research, such as the provided moral classification models, and the number of iterations run.

Provided moral-classification models

A significant limitation of our research is the fact that most of the immoral actions in the game are not detected by the MoralStrength model, thus leading to more overall immorality in the game. Therefore we were compelled to use an oracle instead of the model to still be able to experiment with different configurations. There are several explanations as to why this model fails to detect most moral actions.

Firstly, the moral annotations provided by the environment are annotated by people that have been provided with both the context, as well as the result of the action. The agents, however, do not make use of an oracle, and therefore can only base their prediction on the actual action. One example of an action deemed immoral by the moral annotators is the action "Light candles with match". This is deemed immoral because lighting the candles causes the room to explode, killing the player. The model does not know this, as it does not know the consequence of a potential action.

Secondly, the game often uses ambiguous wording, from which no morality can be predicted. "Take <object>" is an immoral action, since the player steals something. However, this is not deemed to be immoral by the model, since the word "take" is too ambiguous to signal as immoral. This could be solved in the future by explicitly training a model on the game environment, instead of using a model that has been trained on a different domain.

If we compare our research to the Jiminy Cricket research [16], we can see that they use "a RoBERTa-large model [19] fine-tuned on the commonsense morality portion of the ETHICS benchmark", with a 63.4% accuracy. The reason that they do have a functioning morality model is that their model has been trained on a more general domain, the ETHICS benchmark [15], than the models we were provided with that were trained on the Moral Foundations Twitter Corpus [17]. Another reason why their model performs better, is because it is easier to predict whether an action is moral or immoral than to predict which moral value it relates to.

Number of iterations within the timeframe

Another limitation of our research is the number of iterations we could run within the timeframe of this research. Running more iterations would give more valid results and would also show which results were consistent and which were outliers. However, running 1 iteration per experiment would allow us to do more experiments with many different parameters.

7 Responsible Research

In any scientific endeavour, it is crucial to discuss the potential ethical implications of the research, as well as ensure that the methods being used are reproducible. This section will discuss the ethical implications of our research, as well as show the ways we ensure the reproducibility of our research.

7.1 Reproducibility

To make sure that our results are reproducible, we have put all of the code we use on a GitHub repository[1]. This repository contains the Jiminy Cricket environment adapted to use the 5 different moral values, as well as the annotations of the game Suspect.

7.2 Explainability

One important aspect of responsible research is the explainability of Machine Learning, especially in the field of Natural Language Processing. As NLP models continue to advance, the use of 'black-box models' continues to advance as well. These models become increasingly more complex and harder to explain, as is the case with Recurrent Neural Networks or transformer models, often containing many different trained parameters and complex internal representations. This poses several challenges. Firstly, it prevents the detection of data biases [24]. These data biases can emerge from the training data and can lead not only to inaccurate and unfair outcomes but also to systemic prejudices being enforced. Secondly, it is important to understand the reasoning behind the decisions of a model, especially in legal domains, where accountability and trust are paramount. Therefore, explainability is crucial for the ethical development of Machine Learning models.

This is where the MoralStrength model [2] plays a significant role. The MoralStrength model is an explainable model that uses word embeddings to convey the morality of a sentence. The model uses a Moral Foundations Dictionary, where for each lemma a crowd-sourced numeric assessment of Moral Valence is provided, indicating the strength with which a lemma is expressing the specific value. This means that instead of using a black-box model with a sentence as input, the moral valence is given per word, allowing a more critical analysis of the actual returned values per word. This can reveal potential data biases that might otherwise be missed when only looking at the moral valence of the entire input, instead of per word.

7.3 Ethical Implications

Reducing morality to simple parameters will never be without risk. Especially in the case of the Moral Foundations Theory, many actions can be interpreted in different ways, and can therefore have different interpretations according to different people. While we attempted to annotate the actions according to existing psychological research, we are not psychologists, nor do we represent different moral norms and philosophical perspectives. We are all university students sharing similar backgrounds, and can therefore have vastly different annotations compared to any other group from around the globe. To that end, we firmly endorse efforts that can help expand our framework and provide different perspectives.

8 Conclusions and Future Work

This section will conclude our research and will also provide suggestions for future work.

8.1 Conclusion

To conclude our research, we must first refer back to our original research question: *If we one-hot encode the vector, what is the most optimal configuration that can be achieved that maximizes both progress and morality?*. If we only look at the relative immorality of the agent, then running an argmax agent with a care one-hot encoding has resulted in the lowest relative immorality. We can also see that running the agent with argmax often leads to a lower relative immorality. However, this is based on the results of only one-hot encoding the moral values. This will never lead to the most optimal configuration, but rather from the results we can deduce what the most optimal configuration could be. Here, optimal means a configuration that maximizes both progress and morality. How exactly would that translate to the game itself? This would mean that when playing the game, the agent only chooses immoral actions that are absolutely necessary for the progress of the game, and rejects immoral actions that do not contribute to the progress.

From the results we can see that enforcing strict moral boundaries only significantly increases the percent completion in care and purity. This means that using $w_i = (1, 0, 0, 0, 1)$ will not impede the progress of the game. Then the choice of the other weights, w_2, w_3,w_4 will depend on how much either progress or morality should be prioritized, and whether impeding the progress is preferred if it leads to a reduced number of immoral actions.

To test this hypothesis, we also ran an experiment with argmax, w set to 1 and $w_i = (0.5, 0, 0, 0, 0.5)$. We normalized this vector to make sure that the influence of morality in relation to progress remains the same as in the other experiments. Running the experiment results in an immorality of 2.88, a percent completion of 3.78 and a relative immorality of 0.76. Notably, all of these three metrics are the average of the care and purity metrics, showing that results from one-hot encoding the values can be generalisable to other configurations.

Based on these results, we know that using the Moral Foundations Theory will allow for the fine-tuning of morality conditions, and will make it easier to make the distinction between actions necessary for the game, and actions that are superfluous.

[1] https://github.com/enricoliscio/jiminy_cricket_MFTC

8.2 Future Work

This section will describe the recommendations for future work.

Policy shaping vs Reward Shaping There are two different ways to control the behaviour of Reinforcement Learning Agents: Policy shaping and Reward shaping. Reward shaping means that we modify the reward function and policy shaping means that we use other methods than modifying the reward function.

Currently, we use policy shaping to condition the agents towards moral behaviour. We feed the action into a Q-learning network, and after the Q-value per action is calculated, we add a morality parameter to the Q-value. However, by conditioning the agents this way, there is no learning involved. That is because what is fed to the Deep Reinforcement Relevance Network is only how the action causes more progress and in-game rewards, and not how moral the action is. This means that morality is only accounted for in the short term when it is added as a parameter, but not for the long term since the morality parameter is not used for training.

By implementing Reward Shaping as a way to morally condition the agents, the overall morality, as well as the completion percentage, may increase. This could be done by using Q-learning for both progress and morality.

Finding an optimal configuration

Instead of restricting $w_i = (w_1, w_2, w_3, w_4, w_5)$ to be a one-hot vector, we can allow the w_i to take any value between 0 and 1. This would change the problem to a quantifiable problem, finding the optimal configuration for which no higher completion rate can be reached without increasing immorality. It would be feasible to use optimization algorithms such as genetic or local descent algorithms to find the perfect set of weights for w_i.

GPT-3

Right now we use CALM to generate the action candidates per given context. To generate the actions, CALM uses the GPT-2 language model, which at the time of publication in 2019 [22], was the state-of-the-art model. If we set the number of action candidates generated by the model to 40, about 40% of those actions are admissible by the game, and there is an 80% chance that the "gold" action is in this set [25]. This means that if there is an optimal trajectory $(o_1, a_1, \cdots, o_n, a_n)$, and the context c_t is (o_{t-1}, a_{t-1}, o_t) then the *gold* action is a_t However, as of right now, the latest GPT model is the GPT-3.5 model. Using the newest model might increase the number of admissible actions, as well as increase the probability of a "gold" action being in the action candidates set generated by CALM.

Preventing loops

While the argmax agents perform on average better than the softmax agents, they are more unstable because they are bound to get stuck in loops which can greatly increase the immorality as well as impede the percent completion by not progressing in the game. This could easily be prevented by implementing a safeguard in the agent, which would choose another action once an action has been repeated a certain number of times.

References

[1] Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 298–306, New York, NY, USA, 2021. Association for Computing Machinery.

[2] Oscar Araque, Lorenzo Gatti, and Kyriaki Kalimeri. Moralstrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction. *CoRR*, abs/1904.08314, 2019.

[3] Chelsea Barabas, Colin Doyle, JB Rubinovitz, and Karthik Dinakar. Studying up: Reorienting the study of algorithmic fairness around issues of power. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 167–176, New York, NY, USA, 2020. Association for Computing Machinery.

[4] Y. Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. volume 3, pages 932–938, 01 2000.

[5] Elena Boldyreva. Cambridge analytica: Ethics and online manipulation with decision-making process. pages 91–102, 12 2018.

[6] Scott Clifford, Vijeth Iyengar, Roberto Cabeza, and Walter Sinnott-Armstrong. Moral foundations vignettes: A standardized stimulus database of scenarios based on moral foundations theory. *Behavior research methods*, 47, 01 2015.

[7] Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew J. Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. Textworld: A learning environment for text-based games. *CoRR*, abs/1806.11532, 2018.

[8] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 1). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[10] Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition, 1999.

[11] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation, 2018.

[12] Jesse Graham, Jonathan Haidt, Sena Koleva, Matt Motyl, Ravi Iyer, Sean P. Wojcik, and Peter H. Ditto. Chapter two - moral foundations theory: The pragmatic validity of moral pluralism. volume 47 of *Advances in Experimental Social Psychology*, pages 55–130. Academic Press, 2013.

[13] Jesse Graham, Jonathan Haidt, and Brian Nosek. Liberals and conservatives rely on different sets of moral foundations. *Journal of personality and social psychology*, 96:1029–46, 06 2009.

[14] Matthew J. Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. Interactive fiction games: A colossal adventure. *CoRR*, abs/1909.05398, 2019.

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning AI with shared human values. *CoRR*, abs/2008.02275, 2020.

[16] Dan Hendrycks, Mantas Mazeika, Andy Zou, Sahil Patel, Christine Zhu, Jesus Navarro, Dawn Song, Bo Li, and Jacob Steinhardt. What would jiminy cricket do? towards agents that behave morally, 2022.

[17] Joseph Hoover, G J Portillo-Wightman, Leigh Yeh, Shreya Havaldar, Aida Mostafazadeh Davani, Ying Lin, Brendan Kennedy, Mohammad Atari, Zahra Kamel, Madelyn Mendlen, Gabriela Moreno, Christina Park, Tingyee Chang, Jenna Chin, Christian Leong, Jun Leung, Arineh Mirinjian, and Morteza Dehghani. Moral foundations twitter corpus: A collection of 35k tweets annotated for moral sentiment. 04 2019.

[18] John Hutchins. The first public demonstration of machine translation : the georgetown-ibm system , 7 th january 1954. 2006.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[20] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995.

[21] OpenAI. Gpt-4 technical report, 2023.

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[23] Catalin Ungurean and Dragos Burileanu. An advanced nlp framework for high-quality text-to-speech synthesis. In *2011 6th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 1–6, 2011.

[24] Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. A causal view of entity bias in (large) language models, 2023.

[25] Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. Keep calm and explore: Language models for action generation in text-based games, 2020.