Benchmarking Imitation and Reinforcement Learning for serious language-oriented video games

Gema Parreno Piqueras*
Mempathy Author

Madrid
gema.parreno.piqueras@gmail.com

Abstract

This work aims to present the lessons learned of two techniques for solving the NPC behavior in serious language-oriented games in a Reinforcement Learning discrete and partially observable environment. It might be useful as it offers an example of designing companionship in NPCs from the game perspective and presents results for implementing machine learning in NPC players, showing that a designed heuristic function and Imitation Learning approach can speed up developments with respect to a Reinforcement Learning approach for a deterministic output.

1 The video game Mempathy

Mempathy [1] is a video game narrative experience that transforms the relationship with anxiety. The video game's goal is to offer a reflective experience, and the winning state is defined by a feeling of advancement and companionship towards this mental health topic. The idea of progress is supported in art by watercolor progression and in gameplay by discovering a personalized conversation across the different chapters of the game.

1.1 Game Design

The gameplay is developed according to the following structure: firstly, the player unlocks a conversation through clickable objects following a series of blue watercolor scenes making several choices corresponding to several constellations drawn. Secondly, the NPC acts as a companion and is able to respond to the player depending on the player's choice, using a similar mechanic as the player does.

1.2 Designing Companionship for language-oriented games

The NPC develops itself under two principles [2] that help the development of the character through the game and its interaction with the player: as the first principle, the one of *personhood*, defined as the overall impression that the NPC is an independent person inside the game, is reflected in the video game by the NPC having its motivations towards the player (offer encouragement, acceptance, and empathy), with the presence of animated eyes inside the game, and using the same gameplay of the player for guiding the conversation. The second principle is *bonding*: as shared experiences build a deep sense of connection, one of the game's main objectives is to create a bound between the player and the NPC. One of the key challenges here is to overcome some of the factors that could entail a lower bounding, such as superficial and incoherent response or repetitive dialogue. Therefore, the right choice of machine learning techniques in this area has been vital: Reinforcement Learning techniques are oriented towards a specific goal that serves as a motivation for the NPC from the game

*https://github.com/SoyGema

Wordplay: When Language Meets Games Workshop @ NeurIPS 2020, (urlhttps://wordplay-workshop.github.io/

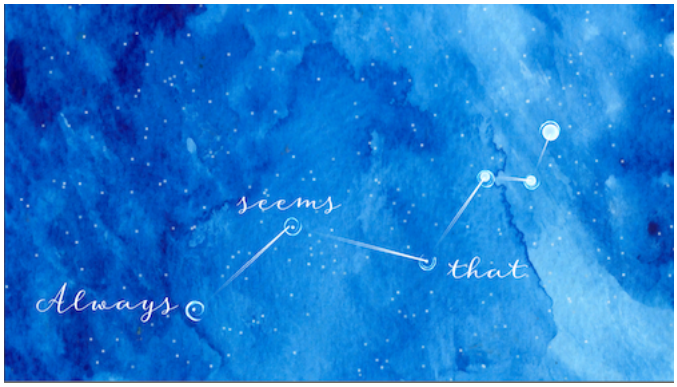

Figure 1: Capture of Mempathy video game. Both the player and the NPC unlock a conversation by clicking on the stars represented as spheres that unlock a conversation in between them.

design perspective. Imitation Learning has been chosen as well as it offers a manner to control the possible NPC's response.

2 Reinforcement Learning Environment

Reinforcement learning is a kind of machine learning method which objective is to maximize the expected discounted cumulative reward, and where the agent learns the optimal policy by trial and error. [3] Considering a discounted episodic Markov decision process (MDP) defined as a tuple (S, A, *y*, P, r), where S is the state space, A is the action space , *y* refers to the discount rate (the present value to apply to future rewards) , the agent chooses an action at according to the policy pi($a_t|s_t$) at state s_t. The environment receives the action, produces a reward r_{t+1} = R(s_t, a_t, s_{t+1}), and transits to the next state s_{t+1} according to the transition probability P($s_{t+1}|s_t$, a_t).

Environments are simulated worlds in which the agent takes actions to reach an specific goal. In Mempathy, the objective is to select words based on player's previous word choices to maximize player's reduction of anxiety. The observation is based on the *n-gram structure* of choosing the word and the *grammatical structure* of the word. The action is based on *choosing the grammatical structure* of the word, the *n-gram prediction* and on *discovering* the word to the player.

Mempathy is a discrete partially observable environment: at each episode, the agent can click on a series of game objects represented as spheres called StarObjects. Each StarObject has a property attached to the game object corresponding with the word's grammatical structure. Each grammatical structure is connected to a database that contains a list of words. The episode terminates when the agent has clicked in all the stars.

At each timestep t, the agent receives the observation matrix. Each row corresponds to a StarObject and each column is based on the n-gram structure of choosing the word (Phase 1) and the grammatical structure of the word (Phase 2) for each StarObject . This two phases create the final observation matrix that the agent will process . This entails that if the agent wants to predict the word W_i , $W_{i-(n-1)}$ has been predicted at timestep t-1 based on $W_{i-(n-2)}$, ..., W_{i-1}) or in probability terms P($W_{i-(n-1)}$ | $W_{i-(n-2)}$, ..., W_{i-1}) where n corresponds to number of words and StarObjects (denoted in figure 2 and created by LookPreviousWord() function). The observation matrix is formed in phase 2 according to the position that must hold inside the vector, depending if the StarObject corresponds to a Noun, Verb, Adjective, Preposition or Adverb in a 1x5 form. The agent then takes an action based on the grammatical structure of the word and the n-gram structure, as it selects the corresponding StarObject and predicts W_i based on $W_{i-(n-1)}$, ..., W_{i-1} or in probability terms P(W_i | $W_{i-(n-1)}$, ..., W_{i-1}) during phase 1 and then clicks on the StarObject discovering the word to the player in phase 2 . For the protoype construction , W_1 has been settled deterministically with an open adverb or noun.

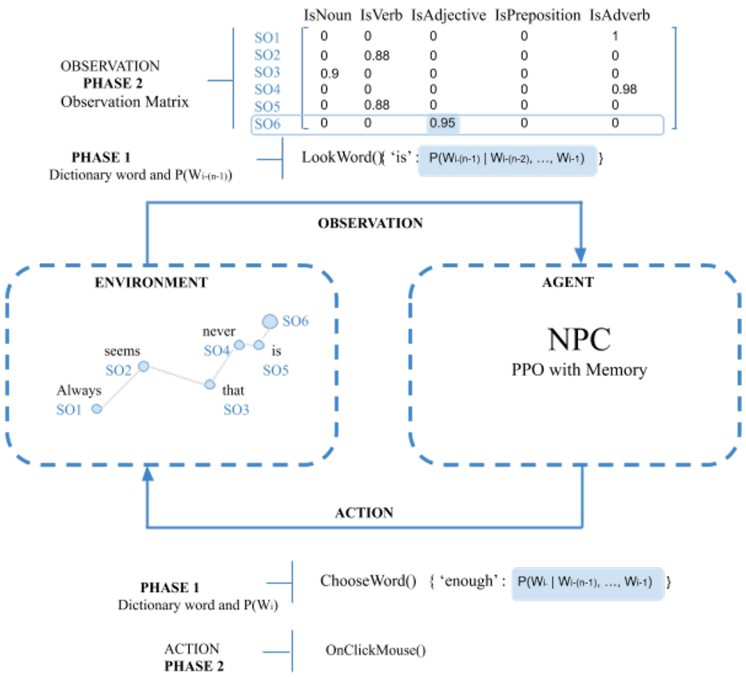

Figure 2: As an example coming from figure 2, as we are observing W_6, the vector of observations for that StarObject will have the form of [0 0 P(W_5 | W_1, ..., W_4) 0 0] where the thrid position stands for the StarObject being an adjective and the P value from the result of the n-gram prediction based on the previous word. The agent will then predict the n-gram probability P(W_6 | W_1, ..., W_5) (enough) and will click on the StarObject, discovering the word to the player

2.1 Reinforcement Learning and Imitation Learning

From a general overview, both Reinforcement and Imitation Learning are methods for sequential tasks, where the agent comes up with a policy in order to achieve the optimal performance. The difference, however, is that in Imitation learning, the agent first observes the actions of an expert during the training phase. The agent uses this training set to learn a policy that tries to mimic the actions demonstrated by the expert, in order to achieve the best performance. In Reinforcement Learning there is no such expert and the agent has a reward function, and it explores the action space for coming up by itself (using trial and error) with an optimal policy.

2.1.1 The Reinforcement Learning experiment

PPO [4] is a policy gradient method for Reinforcement Learning, which alternates between sampling data from the environment and optimizing a surrogate objective function using stochastic gradient ascent. The innovation coming from this method proposes the gradient update in multiple epochs of minibatch updates. This method improves the computational performance and learning stability of Reinforcement Learning implementations.

Reward design is one of the challenges and more fascinating areas of Reinforcement learning [5] as it defines the goal and ultimately is going to shape the agent's behavior. The reward has been designed in a two-fold structure: on the one hand, it rewards the agent for producing sentences with coherent grammatical structure (e.g.: noun, verb, adverb, noun, verb, adverb). On the other, a higher reward is given if the agent chooses in between certain words inside the grammatical structure that correspond with a specific type of sentences aligned with NPC motivations across the scenes that matches NPC's emotional states, acting from the game perspective as a sort of emotion-driven reward response.

As the reward sign given to the agent in this experiment, a 5.0 value has been given everytime the agent clicked on a coherent sequence of the grammatical structure for creating a sequence, and 5.0 more when the agent ends the episode correctly. Besides, if the agent uses certain kind of words in certain scenes, at the end of the episode is given another reward of 5.0. A penalty of 0.5 was given everytime the agent clicked on the wrong StarObject. This entails that for the experiment scenes of 6 words, the maximum cumulative reward per episode might be 40.0 .

For finding the optimal agent, 23 experiments with 2M max steps where trained with tuned batch size order of 10x and buffer size order of 100x, until finding one with a batch size of 64 and a buffer size of 640 that was showing relatively good behaviour. Another 8 experiments with PPO with memory where trained with different sequence lengths of 8 and 16 for saving its experience into memory. Once the memory has this number of experiences, the agent updates its networks using all the experience for 3 epochs. As the NN architecture a 2 layer with 128 hidden units per layer were used for the experiment.

2.1.2 The Imitation Learning experiment

Imitation Learning is based on learning from demonstrations. It uses a system based on the interaction between a *teacher* that performs the task and a *student* that imitates the teacher. In the case of Unity and Unity ML-agents [7], software that has been used for the experiments, the software offers a demonstration recorder where the human acts a teacher, providing examples using the demonstration recorder. Some variants of Imitation Learning, like behavioral cloning, do not use a reward In this case, GAIL [8] algorithm does, the variation of Inverse Reinforcement Learning chosen for the experiment. In Imitation Learning, the set of experiences regarding words has had a significant weight in the results; therefore only coherent sentences across the two levels of observations were trained. Two hundred of demonstrations per scene has been taken.

As the reward given to the agent in this experiment, a reward sign of 5.0 has been given everytime the agent clicked on a coherent sequence of the grammatical structure for creating a sequence, and a penalty of 0.5 was given everytime the agent clicked on the wrong StarObject. This entails that for the experiment scenes of 6 words, the maximum cumulative reward per episode might be 30.0 .

For finding the optimal agent ,12 experiments with 2M max steps where trained with several tuned batch size order of 100x and buffer size with the order of 1000x, until finding one with batch size of 128 and buffer size of 2048 As the NN architecture a 2 layer with 512 hidden units per layer were used for the experiment.

3 Results and training

The best agent is not necesarly defined by solving the episode faster but in showing a optimal behaviour aligned with the objective of the NPC and the game. In this case, the RL agent showed certain pause in clicking and discovering certain words in adjetives and adverbs , which could be interesting from the game design perspective. Both agents solved the tasks showing timing accorded to the gameplay concieved for Mempathy.

If the output aims to be deterministic, training an Imitation Learning agent has been proven better in terms of giving faster results. Therefore , this agent might be recommended under these cincurstances. However, the Reinfocement Learning agent showed an interesting behaviour in showing the adjetives and adverbs that might give a point of view that is more suitable in the long term for thinking about NPC behaviour.

4 Conclusions and future developments

With respect to the game design perspective, the principles of personhood, bonding, and value can offer essential hints for designing NPCs under the goal of creating companionship. Regarding environment design, language can offer an opportunity to use Reinforcement Learning techniques that align with the Reinforcement Learning challenges such as large space complexity and sequence dependence problem.

Regarding reward design inside Reinforcement Learning, the future stands to design directly towards a fully emotion-driven coming from the NPC player motivation. Imitation learning shows faster

activatable results and desirable and controlled behavior during play. If we want a deterministic output, Imitation Learning can significantly speed up video game construction. Besides, using Imitation Learning, the video game industry could introduce in future developments Human in the Loop techniques or players as teachers, designing more personalized experiences for games.

Broader Impact

Even thought Mempathy is not a treatment, as a videogame, could offer a beneficial impact of 3.6 percent of the population that suffers from anxiety disorders [9] , accelerating the path to treatment. From the design perspective , these kind of proposals should have a point of discussion in between the deterministic or stochastic output of the system, and this work might offer an insightfull useful benchmark from the machine learnig perspective.

Acknowledgements

The author would like to thank Jorge Barroso and Beatriz Alonso for the support regarding the technical development and game design fundamentals for the project. A special mention fairly comes to Alberto Hernandez Marcos - BBVA Innovation Labs - for the review and help towards the presented work in this paper.

Thanks to Alexander Zacherl for providing insightfull fundamental documentation about companionship in NPCs , to Maria Dolores Lozano Jimena for the help with syntactic language towards the agents and to Wojciech Czarnecki for testing and providing ideas and feedback.

Special thanks to friends and family for trying out the game and for the encouragement provided by EXAG 2020 and Ladies of Code London communities

References

[1] Mempathy Demo `https://soygema.itch.io/mempathy`

[2] Hiwiller, Z. & Sail, F. (2018) Group Report: Designing Feelings of Companionship with Non-Player Characters The Thirteenth Annual Game Design Think Tank Project Horseshoe .

[3] Shao, K. Tang, Z. Zhu, Y. Li, N. Zhao, D. (2019) A survey of Deep Reinforcement Learning in Video Games. IEEE

[4] Schulman, J. et al. (2017) Proximal Policy Optimization Algorithms.

[5]Sutton, R. & Barto G. Reinforcement Learning: An Introduction.

[6] Zuo, S. Wang, Z. (2017) Continuous Reinforcement Learning from Human Demonstrations with Integrated Experience replay for Autonomous Driving.

[7] Juliani, A. et al. 2020 Unity: A General Platform for Intelligent Agents.

[8] Ho, J. & Ermon, S. (2016) Generative Adversarial Imitation Learning

[9] World Health Organization (2017). Depression and Other Common Mental Disorders.

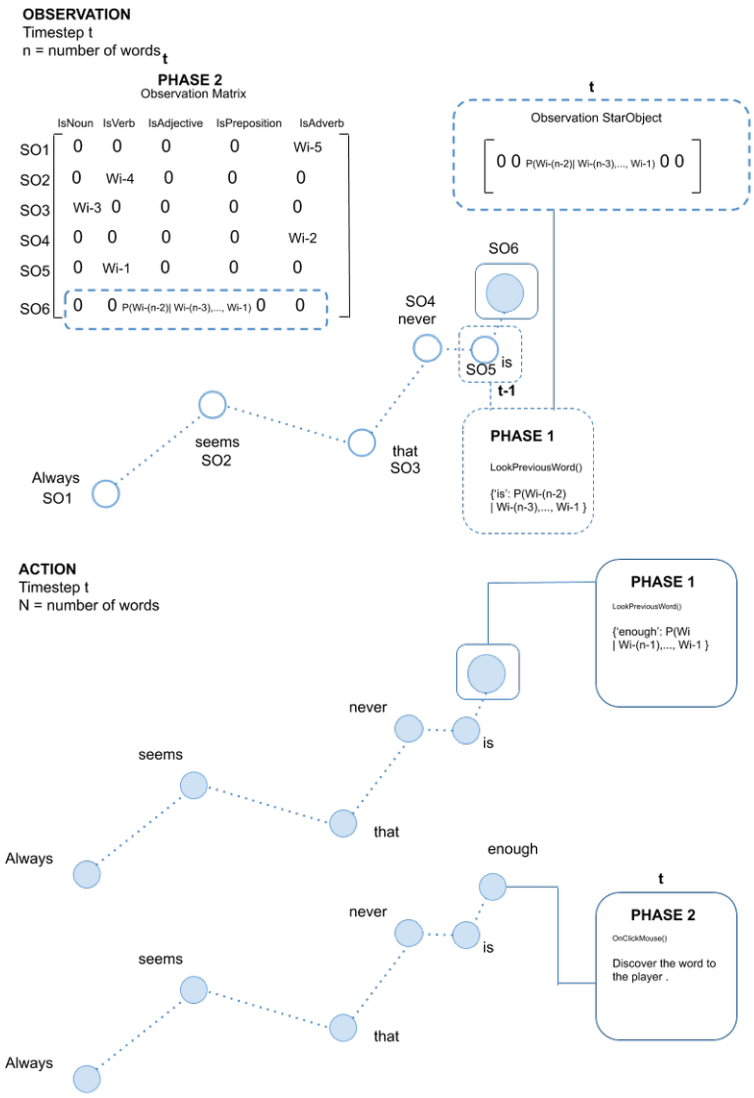

Figure 3: Diagram showing Observation and Actions of StarObject n = 6. When observing this StarObject, the agent looks at the n-gram probability distribution of StarObject number 5 and introduces it into the vector of observations corresponding with the grammatical structure of the StarObject. The final vector of observations for each StarObject includes the syntatic structure and the n-gram structure

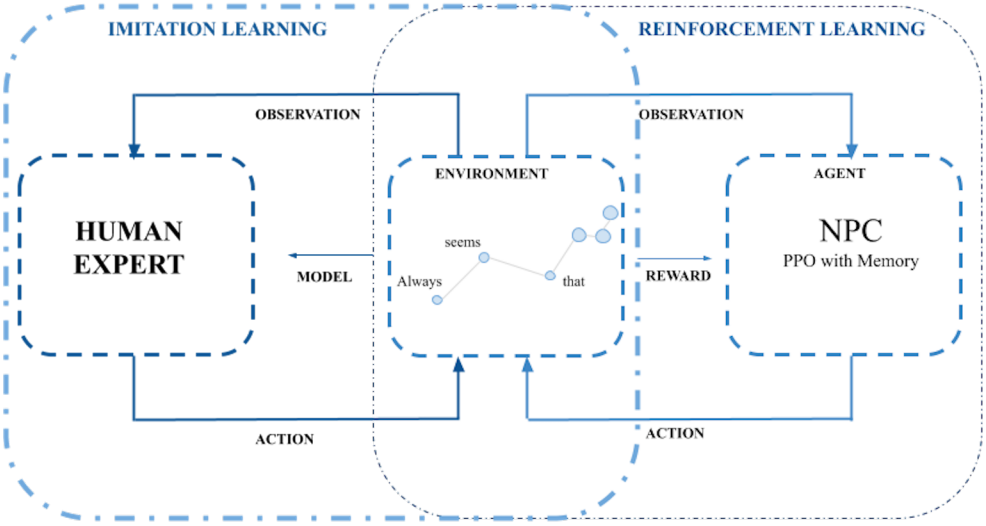

Figure 4: Diagram showing how the learning loop works in Reinforcement and Imitation Learning. Both share Mempathy environment and the actions are provided by a human [6]

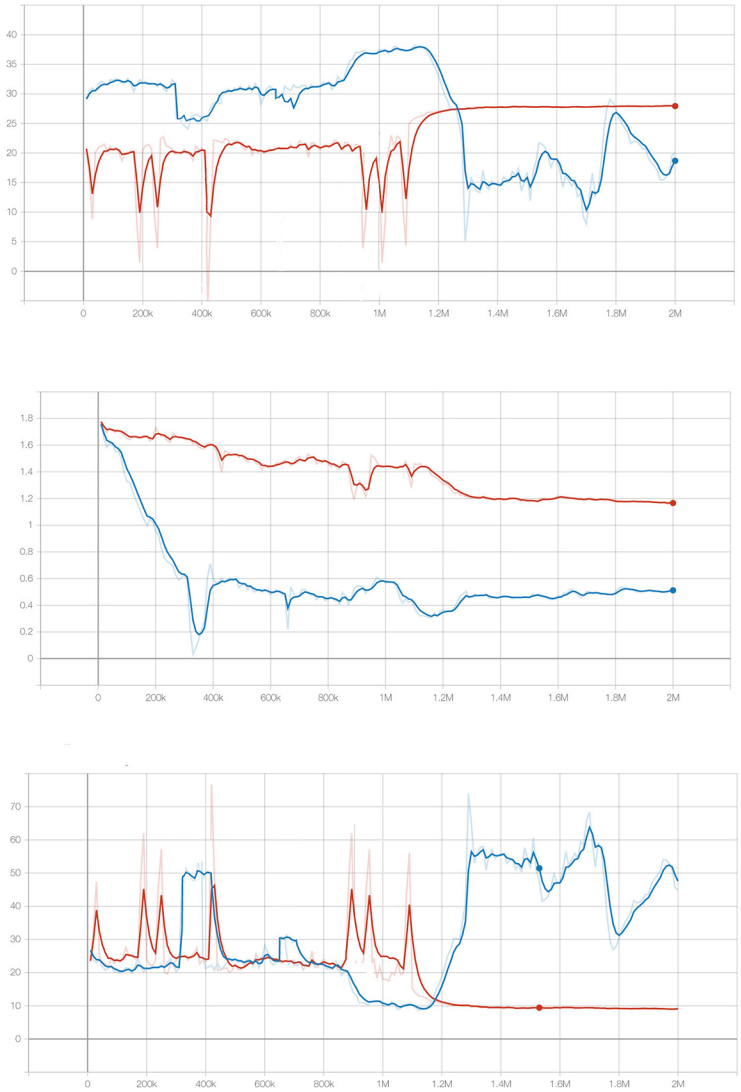

Figure 5: Comparisson of the two best performing agents in terms of behavior in PPO (blue) VS GAIL(red) in cumulative reward, entropy and episode length . The GAIL agent gets closer to its maximum reward after 1M training steps . Besides, RL agent shows lower entropy , which entails less diversity in the actions chosen by the policy.