VILNIUS TECH
Faculty of Fundamental Sciences

Reinforcement Learning Based Model for Personalising the Picture Exchange Communication System

Asta Slotkienė, Augustas Mikulėnas

Abstract. The ability to communicate effectively with children diagnosed with autism spectrum disorders (ASD) is an essential skill for adults. To facilitate communication, researchers have proposed augmentative and alternative communication systems, such as the Picture Exchange Communication System (PECS). To improve their usability, the systems are digitalized. Artificial Intelligence (AI) capabilities are applied to increase communication effectiveness. Most existing research works based on AI models try to build classifiers for the child's reaction states, assuming that the data to train the models are fully labeled. However, data labeling is prone to subjective interpretations by the specialist. For each child with ASD, their individuality makes it impossible to determine the rule-based data and their decisions. In this paper, we propose a strategy learning model, where the agent uses reinforcement learning (RL) to learn an optimal strategy of presenting an appropriate PECS card for effective communication between a child with ASD and an adult. The study included experiments with different RL environments and their parameters, describing different sets of states and prescribing additional metadata to find the optimal learning strategy. The main result is the proposed new RL-based model, which allows the achievement of the goal with a minimal number of steps by reducing the count of episodes and utilizing a strategy that maximizes the probability of achieving the goal.

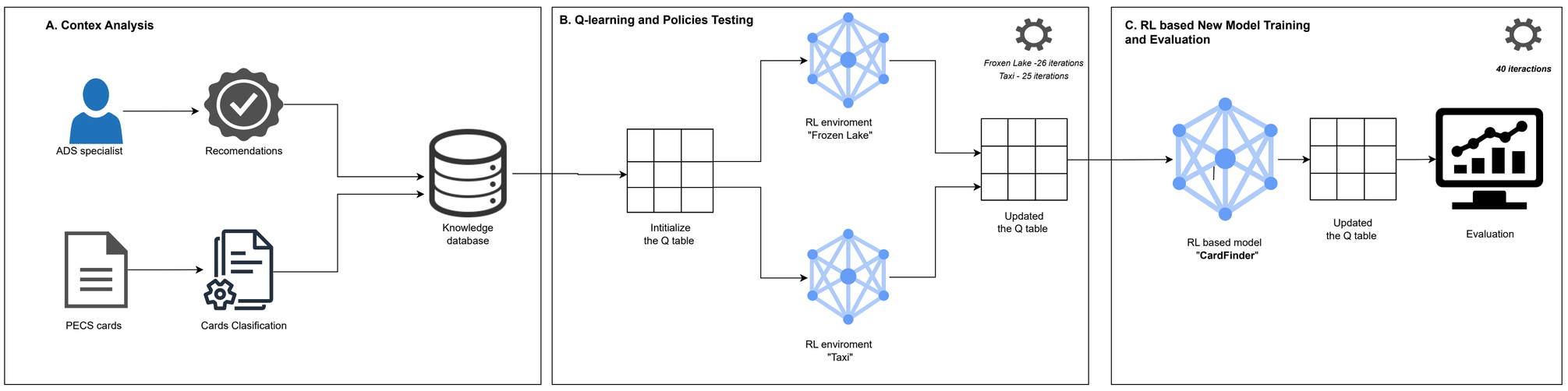

Figure 1. The principal scheme of the performed research

A. Context Analysis.

Data collecting and data analysis were prepared with the help of VšĮ Vilnius University Hospital Santaros Clinics Child Development Center specialists, who work with children having autism. PECS cards were applied for the study. According to specialists, it is impossible to develop the same algorithm for improving communication for all children. 81 cards classified into 9 categories were used for the research. The specialist recommended creating relationships between cards, that reduce mistakes and would not cause a child's negative reaction.

B. Q-learning and Policies Testing Process

The Q-learning Testing Process involves Q value functions to estimate how good it is to perform an action in a certain state. This action-value function returns the expected cumulative reward of a sequence of actions which starts from action a_t in state s_t and thereafter follows policy π:

$$Q^{\pi}(s_t, a_t) = \sum[R_t | s_t, a_t] \text{ [1]}$$

With the Bellman equation, we express the relationship between the value of a state-action pair and its successor state-action pair:

$$Q^{\pi}(s_t, a_t) = \sum[r_t(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, \pi_{s_{t+1}})] \text{ [2]}$$

The Q-learning uses an equation (see 2 equation) to estimate each state-action pair. If the states and actions are discrete and finite, the pairs can be represented in a tabular form where all pairs will be given an initial value. Every time an action is executed, the related state-action value will be updated:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma Q^{*}(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \text{ [3]}$$

$Q*(s_{t+1},a_{t+1})$ represents the maximum cumulative reward that can be achieved from state s_{t+1}.

For this, we apply two different RL environments (Frozen Lake, Taxi), which helps us find effective policy algorithms.

C. RL based New Model Training and Evaluation

```
set: learningRate, gamma, epsilon, maxEpsilon, minEpsilon, decayRate, totalEpisodes, maxSteps
set negativeReward, positiveReward, stepsReward, verypositiveReward

enviroment.generateMaps(size, probabilityofCardisNeutral, probabilityOfCard):
map = randomChoice (["NeutralCard", "PositiveCard", "NegativeCard"]
set "StartCard" ir "LastCard"
jumpers = randomAssignCard()

while episode in totalEpisodes:
  enviroment.reset()

  while step < maxSteps:
    generate RandomValue
    if RandomValue > epsilon:
    action = Qtable[state]
    else:
     action = random.actionList
     newState, reward = enviroment.step(action):
    if action == LEFT:
         agent move to the left
    elif action == DOWN:
        agent move to the down
    elif action == RIGHT:
        agent move to the right
    elif action == UP:
        agent move to the top
    elif action == META:
         agent move to the by metavalue.
   Qtable[state,action]=qtable[state,action]+learningRate*(reward+gamma*Qtable[newstate,])-qtable[state, action])
   if goalAchieved :
      stop episodes

  reduce epsilon value
   epsilon = min_epsilon + (maxEpsilon - minEpsilon)*np.exp(-decayRate*episode)
```

Figure 2. Pseudo code of the CardFinder environment

For RL-based model training, 40 iterations of different parameters were performed with the suggested CardFinder environment. Experimental result shows, that the agent has positive feedback, when we reduce the probability to even 5% (to show positive or very positive cards), even in cases where 81 possible states exist, the goal is achieved in less than 5 steps. It was also found that in the suggested CardFinder environment the agent is able to learn in an extremely low number of episodes, whereas when testing with 81 available cards, even after reducing the number of episodes to 15, the agent still reached the goal in approximately 5 steps.

Conclusion

We suggested a new RL-based model, which allows for children with ADS to achieve the goal during the five steps by reducing the count of episodes until 15 and utilizing a strategy that maximizes the probability of achieving the goal - to show the PECS card, which helps for the child with ADS to express the mind and wishes. The chosen exploratory way has been proven because it is usually impossible to describe and label the initial data set when each child has different needs and contexts of communication.