Reinforcement Learning Strategies for Pecs Card Selection in Augmentative and Alternative Communication Systems

Paulina Ivanauskaitė, Asta Slotkienė

Vilnius University, Faculty of Mathematics and Informatics,
Institute of Computer Science,
Naugarduko st. 24, LT-03225, Vilnius, Lithuania
paulina.ivanauskaite@mif.vu.lt

Abstract. Autism spectrum disorder affects communication abilities, making augmentative and alternative communication systems essential for nonverbal individuals. The Picture Exchange Communication System is a structured, symbol-based method that enables nonverbal individuals to initiate communication by exchanging picture cards representing desired objects, actions, or concepts. It is widely used but requires efficient card selection strategies to maximize successful exchanges. This research evaluates the effectiveness of reinforcement learning algorithms combined with different card selection strategies. The experiments revealed that introducing opposite card selection after rejection significantly improved success rate compared to similarity-only approaches. The findings demonstrate that strategy design has greater impact on system performance than the choice of learning algorithm.

Keywords: reinforcement learning, PECS, AAC, autism, Q-learning, card selection strategies.

1 Introduction

Autism Spectrum Disorder (ASD) is a neurodevelopmental condition characterized by difficulties in social interaction, communication, and repetitive behaviours [1]. Augmentative and Alternative Communication (AAC) systems provide essential support for individuals with limited verbal abilities [1]. Among these systems, the Picture Exchange Communication System (PECS) has gained widespread adoption due to its structured approach and visual nature [2]. PECS enables users to communicate by exchanging picture cards representing objects, actions, or concepts. A critical challenge in PECS implementation is efficient card selection - determining which cards to present to the user at any given moment. Traditional

DOI: https://doi.org/10.15388/LMITT.2026.9

approaches rely on predetermined sequences or therapist intuition, which may not adapt well to individual user preferences and learning patterns. This paper demonstrates the research of the effectiveness of different card selection strategies when combined with reinforcement learning (RL) algorithms and their parameters.

The remainder of this paper is structured as follows. Section 2 presents the proposed research methodology. The experiment results are presented in Section 3. Conclusions are drawn in Section 4.

2 Research methodology

The research methodology consists of eight sequential phases illustrated in Fig. 1: the research data preparation, RL algorithm selection, the configuration of PECS card selection strategy, exploration method selection, similarity score calculation method selection, virtual child configuration, experiment execution, and result analysis.

The research data preparation phase (Fig. 1, step 1) involves encoding PECS cards as feature vectors. The cards database contains 17 fruit items, each described by six features (0-1 scale): hardness, sweetness, sourness, shape, texture, and colour. These features enable similarity calculations between cards. For example, Banana has features (0.3, 0.9, 0.1, 1.0, 0.0, 0.33), Apple has (0.8, 0.7, 0.4, 0.0, 0.2, 0.0), and Mango has (0.5, 0.9, 0.3, 0.2, 0.1, 0.25). Table 1 presents selected items from the database.

Table 1. Sample of fruit features values.

Product	Hardness	Sweetness	Sourness	Shape	Texture	Color
Banana	0.3	0.9	0.1	1.0	0.0	0.33
Apple	0.8	0.7	0.4	0.0	0.2	0.0
Mango	0.5	0.9	0.3	0.2	0.1	0.25

Two similarity scores were implemented to compare card feature vectors (Fig. 1, step 5). Cosine similarity score [4] measures the angle between two vectors *A* and *B* (Equation 1):

$$\cos(\mathrm{A}, \mathrm{B}) = \frac{A \cdot B}{\|A\| \times \|B\|}. \quad (1)$$

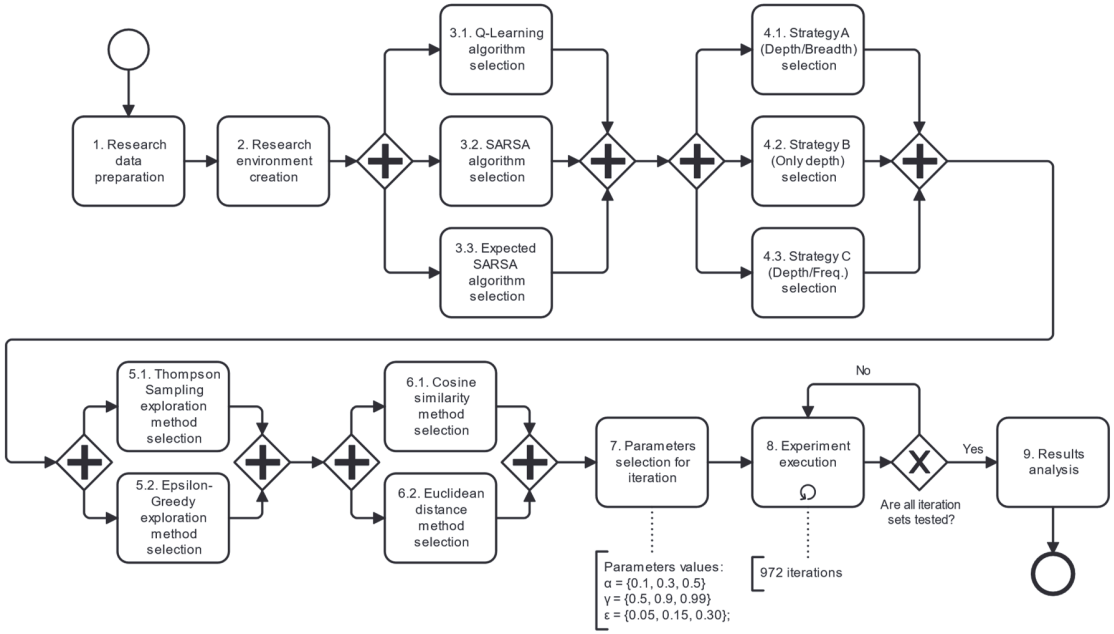

Fig. 1. Research methodology pipeline showing the eight phases from data preparation to result analysis.

In the equation *A* and *B* are feature vectors representing two PECS cards being compared. Each vector has 6 components ($n = 6$), corresponding to the features (Table 1). $A \cdot B$ is the dot product and $||A||$, $||B||$ are vector magnitudes. The result *cos(A, B)* ranges from 0 (orthogonal) to 1 (identical direction).

Euclidean similarity score [4] is derived from Euclidean distance (Equation 2):

$$\mathrm{sim}(\mathrm{A}, \mathrm{B}) = \frac{1}{1 + \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}}. \quad (2)$$

The result *sim(A, B)* is Euclidean distance converted to a 0-1 similarity scale where 1 indicates identical items. Here, A_i and B_i denote the i-th feature components of vectors *A* and *B*, and *n* is the total number of features ($n = 6$ in this research).

The RL algorithm selection phase (Fig. 1, steps 2.1-2.3) involves choosing from three temporal-difference (TD) learning algorithms: Q-learning [5], SARSA [6], and Expected-SARSA [7]. Q-learning is an off-policy algorithm that updates Q-values using the maximum expected future reward. SARSA is an on-policy algorithm that updates Q-values based on the action selected. Expected-SARSA updates Q-values using the expected value over all possible next actions. Two exploration methods were tested (Fig. 1, steps 3.1-3.2): epsilon-greedy (random exploration with probability ε) and Thompson sampling [8] (Bayesian approach using beta distribution sampling).

Based on AAC and ASD behaviour [8], the three card selection strategies were designed (Fig. 1, steps 4.1-4.3). Strategy A (depth + breadth) presents the top two cards according to Q-values on the first attempt. After rejection, it presents one similar card (most like rejected cards) and one opposite card (least like rejected cards). This approach hedges against uncertainty: if the initial guess was close, the similar card may succeed; if it was wrong entirely, the opposite card provides a corrective option.

Algorithm 1. Pseudocode of strategy A (depth + breadth)

```
Input: available_cards, rejected_cards, rejection_count, Q_table
Output: card1, card2
1:  if rejection_count = 0 then
2:      card1, card2 = SelectByExploration(Q_table, available_cards)
3:  else
4:      card1 = MostSimilar(rejected_cards, available_cards)
5:      card2 = MostOpposite(rejected_cards, available_cards)
6:  end if
7:  return card1, card2
```

Strategy B (only depth) also presents top two Q-value cards on the first attempt, but after rejection always selects the two most similar cards. This strategy focuses narrowly on the presumed preference region, assuming the user wants something close to what was rejected. This can be effective when initial guesses are in the right direction but may fail when they are not.

Algorithm 2: Pseudocode of strategy B (only depth)

```
Input: available_cards, rejected_cards, rejection_count, Q_table
Output: card1, card2
1:  if rejection_count = 0 then
2:      card1, card2 = SelectByExploration(Q_table, available_cards)
3:  else
4:      card1, card2 = MostSimilar(rejected_cards, available_cards, n=2)
5:  end if
6:  return card1, card2
```

Strategy C (depth + frequency) adapts based on rejection count. On the first attempt, it presents top two Q-value cards. After the first rejection, it combines one similar card with one Q-value-based card, maintaining some exploration. After multiple rejections, it switches to two similar cards. This gradual narrowing differentiates it from both Strategy A (which always includes opposites) and Strategy B (which immediately narrows).

Algorithm 3: Pseudocode of strategy C (depth + frequency)

```
Input: available_cards, rejected_cards, rejection_count, Q_table
Output: card1, card2
1:  if rejection_count = 0 then
2:      card1, card2 = SelectByExploration(Q_table, available_cards)
3:  else if rejection_count = 1 then
4:      card1 = MostSimilar(rejected_cards, available_cards)
5:      card2 = SelectByExploration(Q_table, available_cards)
6:  else
7:      card1, card2 = MostSimilar(rejected_cards, available_cards, n=2)
8:  end if
9:  return card1, card2
```

The virtual child configuration phase (Fig. 1, step 6) defines a deterministic preference model: when presented with two cards, the virtual child selects their desired item if present, otherwise rejects both cards. The preference sequence consists of 100 predefined wants, with Banana appearing most frequently (40%), followed by Apple, Pear, Mango, Tangerine, Blueberries, and Grapes.

A session is a single communication attempt in which the system presents cards to the virtual child until either the desired card is selected or

the maximum number of attempts (3) is reached. The effectiveness of each configuration is measured by the *success_rate* (Equation 3):

$$\text{success\_rate} = \left(\frac{succesful\_sessions}{total\_sessions}\right) \cdot 100\%. \quad (3)$$

In this equation *successful_sessions* is the number of sessions in which the virtual child selected their desired card within the allowed 3 attempts, and *total_sessions* is the total number of sessions run per experiment (100, corresponding to the 100 predefined wants in the preference sequence). Sessions are grouped into two categories: successful (desired card selected within the attempt limit) and unsuccessful (no selection within the attempt limit).

The experiment execution phase (Fig. 1, step 7) tests all parameters combinations. Each experiment iteration is executed using one specific combination of parameters. A combinations are formed by selecting one value from each parameter: one of the three algorithms is chosen, then one of the three values: α values, γ values, ε values, applied two exploration methods and three determined strategies, and the similarity score was calculated by two different methods (Table 2).

Table 2. The experiment's parameters grid.

Parameter	Values
Algorithm	Q-learning, SARSA, Expected-SARSA
Learning rate (α)	0.1, 0.3, 0.5
Discount factor (γ)	0.5, 0.9, 0.99
Exploration rate (ε)	0.05, 0.15, 0.30
Exploration method	Epsilon-greedy, Thompson sampling
Strategy	A, B, C
Similarity score methods	Cosine, Euclidean

Since each parameter is selected independently of the others, the total number of possible combinations is calculated by multiplying the number of values for each parameter: 3 × 3 × 3 × 3 × 2 × 3 × 2 = 972 unique combinations.

3 Results

The goal of this research was to determine which combination of card selection strategy, RL algorithm, and exploration method achieves the highest success rate in PECS card selection. Table 3 summarizes the success rate statistics grouped by card selection strategy.

Table 3. Success rate statistics by card selection strategy.

Strategy	Mean (%)	Std (%)
A (depth + breadth)	65.40	4.12
B (only depth)	48.90	7.89
C (depth + frequency)	54.40	8.45

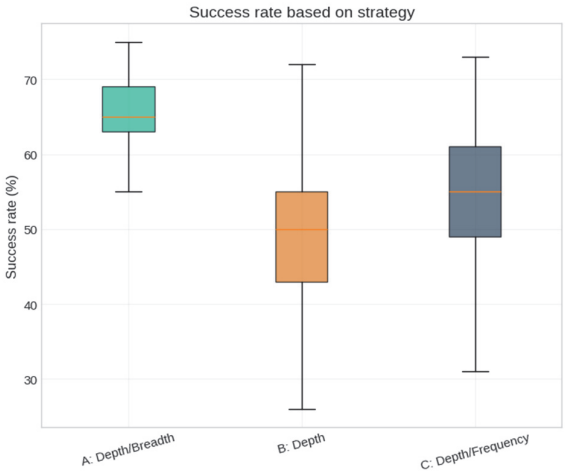

Fig. 2. Success rate distribution by strategy.

Strategy A achieved the highest effectiveness where success score mean was achieved 65.4%, median 66.0%, and standard deviation 7.2%, demonstrating that introducing opposite card selection after rejection significantly improves results compared to similarity-only approaches. Strategy B achieved mean 48.9%, median 49.0%, and standard deviation 5.8%. Strategy C achieved mean 54.4%, median 55.0%, and standard deviation 6.5%. Fig. 2 visualizes success rate distribution by strategy.

Table 4. Success rate statistics by exploration method.

Exploration method	Mean (%)	Std (%)
Thompson sampling	60.35	7.23
Epsilon-greedy	52.15	10.12

As shown in Table 4, Thompson sampling significantly outperformed epsilon-greedy across all configurations, achieving 60.35% mean success rate compared to 52.15% for epsilon-greedy. The 8.2 percentage advantage suggests that the Bayesian approach to exploration-exploitation balancing is better suited for this problem, where uncertainty about user preferences must be efficiently resolved.

Table 5. Success rate by RL algorithm.

Algorithm	Mean (%)	Std (%)
Q-learning	56.08	9.88
SARSA	56.67	10.02
Expected-SARSA	55.98	10.16

Interestingly, the three TD algorithms showed minimal performance differences (less than 1% variation). As illustrated in Table 5, Q-learning achieved 56.08%, SARSA 56.67%, and Expected-SARSA 55.98%. This similarity suggests that card selection strategy and exploration method have greater impact than the specific TD algorithm. Table 6. Success rate by similarity calculation method.

Table 6. Success rate by similarity calculation method.

Similarity method	Mean (%)	Std (%)
Cosine	56.77	9.85
Euclidean	55.73	10.15

According to Table 6, cosine similarity slightly outperformed Euclidean similarity (56.77% vs 55.73%), though the difference of approximately 1 percentage was not statistically significant. This suggests that both methods are viable for PECS card similarity calculations.

As shown in Table 7, the learning rate (α), discount factor (γ), and exploration rate (ε) parameters had minimal impact on success rate, with variations of less than 1 percentage across all tested values. This indicates

that the system is robust to hyperparameter choices within the tested ranges. The best configuration achieved 75.0% success rate with Strategy A, Thompson sampling, learning rate 0.3, discount factor 0.9, and cosine similarity.

Table 7. Success rate by hyperparameter values.

Parameter	Value	Mean (%)	Std (%)
Learning rate (α)	0.1	56.52	9.94
Learning rate (α)	0.3	56.33	10.08
Learning rate (α)	0.5	55.87	10.06
Discount factor (γ)	0.50	56.12	10.15
Discount factor (γ)	0.90	56.19	9.98
Discount factor (γ)	0.99	56.41	9.96
Exploration rate (ε)	0.05	56.48	9.89
Exploration rate (ε)	0.15	56.27	10.04
Exploration rate (ε)	0.30	55.97	10.15

4 Conclusions

Our proposed research methodology allows evaluating effectiveness of AAC systems from several aspects: RL algorithm selection, card selection strategy design, and exploration method choice. The results demonstrate that PECS card selection strategy is the most influential factor. With PECS card strategy A (combining similar and opposite cards after rejection) achieving significantly higher effectiveness than narrow search approaches. This superiority stems from Strategy A's ability to cover a wider preference range per attempt: the similar card succeeds when the initial guess was close, while the opposite card provides a corrective option when it was not. In practical terms, this reduces failed sessions, which is especially important in therapeutic contexts where repeated rejection can cause frustration and disengagement. The second aspect which influenced the success rate was the exploration method, where Thompson sampling demonstrated an 8 percentage advantage over epsilon-greedy.

Thompson sampling provides superior exploration-exploitation balance compared to epsilon-greedy. The Bayesian approach naturally balances exploration and exploitation by sampling from posterior distributions, effectively reducing uncertainty more efficiently than random exploration.

Notably, the specific TD algorithm (Q-learning, SARSA, Expected-SARSA) has minimal impact, suggesting that strategy design should be prioritized over algorithm selection. These findings provide a foundation for developing adaptive PECS systems that can learn individual user preferences and improve communication efficiency over time.

Future work should introduce stochastic child behaviour into the simulation model, replacing the current deterministic preference model with a probabilistic one to better reflect real-world variability in user responses. Additionally, exploring a broader range of card selection strategies beyond the three evaluated in this study represents a promising direction for further research.

References

[1] Shaw, K. A., Williams, S., Patrick, M. E., et al. (2025). Prevalence and Early Identification of Autism Spectrum Disorder Among Children Aged 4 and 8 Years — Autism and Developmental Disabilities Monitoring Network, 16 Sites, United States, 2022. *MMWR Surveillance Summaries*, 74(2), 1-22.

[2] Pope, L., Light, J., & Laubscher, E. (2024). The effect of naturalistic developmental behavioral interventions and aided AAC on the language development of children on the autism spectrum with minimal speech: A systematic review and meta-analysis. *Journal of Autism and Developmental Disorders*, 55(9), 3078-3099.

[3] Bondy, A., & Frost, L. (2002). The picture exchange communication system training manual (3rd ed.). Pyramid Educational Consultants.

[4] Aggarwal, C. C. (2022). Machine learning for text (2nd ed.). Springer.

[5] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

[6] Rummery, G. A., & Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University.

[7] Van Seijen, H., et al. (2009). A theoretical and empirical analysis of expected Sarsa. IEEE ADPRL, 177-184.

[8] Russo, D., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on Thompson sampling. Foundations and Trends in Machine Learning, 11(1), 1-96.

[9] Lepper, T. L., Petursdottir, A. I., & Esch, B. E. (2013). Effects of operant discrimination training on the vocalizations of nonverbal children with autism. *Journal of Applied Behavior Analysis*, 46(4), 509-521.