AI-Powered Food Akinator Using Q-Learning and Information Gain Entropy

Aaryan Darad*, Atharva Tiwari†, Shrijan Sahu‡, Siddhi Rajpurohit§, Smit Mehta¶, Yash Patel‖
IIT Gandhinagar, Gujarat, India
Emails: *dardaaryan@iitgn.ac.in, †tiwariatharva@iitgn.ac.in, ‡sahushrijan@iitgn.ac.in,
§rajpurohitsiddhi@iitgn.ac.in, ¶mehtasmit@iitgn.ac.in, ‖patelyash@iitgn.ac.in

***Abstract*—This report presents the development of the Food Akinator, an AI-powered interactive guessing game. We developed two versions of the game: one leveraging Information Gain Entropy to optimize question selection (Version A) and another based on Q-Learning (Version B). A user-friendly interface was built using Python's `tkinter` library, and a dataset of food items with attributes like vegetarian, spicy, and vegan was employed. With multiple difficulty levels, the game is suitable for all users. The project demonstrates the creative application of reinforcement learning and information-theoretic approaches in interactive systems.**

I. Introduction

The Food Akinator is a guessing game where the AI identifies a food item by asking questions. Inspired by traditional guessing games, it incorporates AI techniques to make the gameplay smarter and more dynamic. Two versions of the game were developed:

- **Version A:** Uses Information Gain Entropy to optimize question selection.
- **Version B:** Uses Q-Learning to learn from user interactions and improve performance over time.

A. Features of Version A

- **Objective:** To guess a food item by dynamically narrowing down possibilities.
- **Interactive Gameplay:** Players respond to AI questions (`Yes`, `No`, `Maybe`) through a graphical interface.
- **AI Adaptability:** The AI decides to ask the question that results in the highest information gain, based on the user's answers.

B. Features of Version B

- **Objective:** To guess a food item by dynamically narrowing down possibilities.
- **Interactive Gameplay:** Players respond to AI questions (`Yes`, `No`, `Maybe`) through a graphical interface.
- **Difficulty Levels:** The game supports:
  - **Easy:** Random feature selection.
  - **Medium:** Prioritizes unasked features.
  - **Hard:** Uses Q-Learning to strategically choose questions.
- **AI Adaptability:** In Hard mode, the AI learns from player interactions and improves its guessing strategy.

II. Methodology

The development of the Food Akinator involved the following steps:

A. Dataset Preparation

- A dictionary of food items with attributes like vegetarian, spicy, and vegan was created.
- The dataset was converted into a pandas DataFrame to facilitate efficient filtering and processing.

B. Graphical User Interface

- Built using Python's `tkinter` library.
- Features include:
  - A display for the Akinator character.
  - Buttons for player responses (`Yes`, `No`, `Maybe`).

C. Version A: Information Gain Entropy Implementation

- **Information Gain Entropy:** This version selects questions based on the concept of information gain. The AI calculates the entropy of possible food items and selects the question that maximizes information gain, thereby reducing the uncertainty most effectively.
- **Entropy Calculation:** The entropy of the remaining food items is calculated, and the question with the highest expected information gain is chosen.
- **Key Steps:**
  - Calculate entropy for each potential question.
  - Select the question that provides the highest reduction in entropy.
  - Filter the dataset based on player responses.

D. Version B: Q-Learning Implementation

- A Q-table initialized with zeros stores Q-values for feature-action pairs.
- The update formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max Q(s',a') - Q(s,a) \right]$$

- Key parameters:
  - `alpha`: Learning rate for updating Q-values.
  - `gamma`: Discount factor for considering future rewards.
  - `reward`: Reduction in dataset size after filtering.

E. Pseudo Code

Below are the pseudo codes of the implementation for Akinator of food items. The actual codes and implementation in detail can be found in the GitHub repository - **daradaaryan/Akinator-F-of-AI**

Algorithm 1 Version A - Information Gain-Based Selection

function CALCENTROPY(probs)
  $H \leftarrow -\sum_{p \in probs} p \cdot \log_2(p)$ where $p > 0$
  return H
end function
function CALCINFOGAIN(feat)
  $yProb \leftarrow \sum_{p(food)}$ where $feat$=True
  $nProb \leftarrow 1 - yProb$
  if $yProb = 0 \vee nProb = 0$ **then**
    return 0
  end if
  $yEnt \leftarrow$ CALCENTROPY($\frac{p(food)}{yProb}$ where $feat$ = True)
  $nEnt \leftarrow$ CALCENTROPY($\frac{p(food)}{nProb}$ where $feat$ = False)
  return $H_{\text{current}} - (yProb \cdot yEnt + nProb \cdot nEnt)$
end function
function SELECTQUESTION(feats, probs)
  $featGains \leftarrow$ CALCINFOGAIN(feat) for all $feat \in feats$
  return $\arg\max_{feat} featGains$
end function

Algorithm 2 Version B - Q-Learning Feature Selection

function SELECTFEAT(feats, askedFeats, qTbl)
  $remFeats \leftarrow feats \setminus askedFeats$
  if $remFeats = \emptyset$ **then**
    return None
  end if
  $qVals \leftarrow$ Sum of Q-values across actions for $remFeats$
  $maxQ \leftarrow \max(qVals)$
  $bestFeats \leftarrow$ features with Q-value $maxQ$
  return Random choice from bestFeats
end function
function UPDATEQ(feat, act, rew, nextFeat, qTbl, α, γ)
  $curQ \leftarrow qTbl[feat, act]$
  $maxNextQ \leftarrow \max(qTbl[nextFeat])$ if $nextFeat$ else 0
  $updQ \leftarrow curQ + \alpha \cdot (rew + \gamma \cdot maxNextQ - curQ)$
  $qTbl[feat, act] \leftarrow updQ$
end function

F. Game Flow

- The player selects a language and difficulty level.
- **Version A**: Strategic selection using Information Gain Entropy.
- **Version B**: The AI selects a question based on the mode:
  - Easy/Medium: Random or prioritized unasked features.
  - Hard: Strategic selection using Q-Learning.
- The dataset is filtered based on player responses.
- The AI makes a final guess once confident.

III. OUTCOME

The Food Akinator achieves:

- Accurate guessing through iterative filtering and questioning.
- Adaptive feature selection in Hard mode via Q-Learning (Version B).
- Optimized question selection using Information Gain Entropy (Version A).
- Multi-language support and adjustable difficulty levels (Version B).

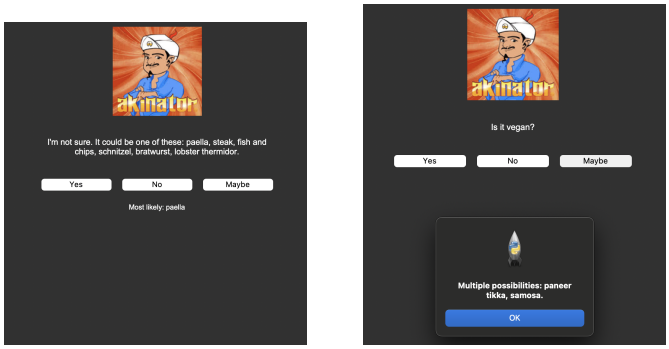

Fig. 1. Version A (left) and Version B (right)

IV. SCOPE OF IMPROVEMENT

Future enhancements for the Food Akinator include:

- **Smarter AI:** Replace Q-Learning with neural networks for better generalization.
- **Free-Text Answers:** Integrate NLP to process typed responses for a more natural interaction.
- **Expanded Dataset:** Add more food items and features for broader applicability.
- **Dynamic Difficulty:** Adjust gameplay difficulty based on player performance.

V. ACKNOWLEDGMENTS

We would like to express our gratitude to Prof. Neeldhara Misra and Prof. Manisha Padala for giving us the opportunity to work on this project and apply the knowledge gained inside the classroom in such innovative ways.

REFERENCES

[1] Elokence. (2024). Akinator, the Web Genius. [Online]. Available: https://en.akinator.com/

[2] Elokence. (2024). About Us. [Online]. Available: https://us.akinator.com/content/7/about-us