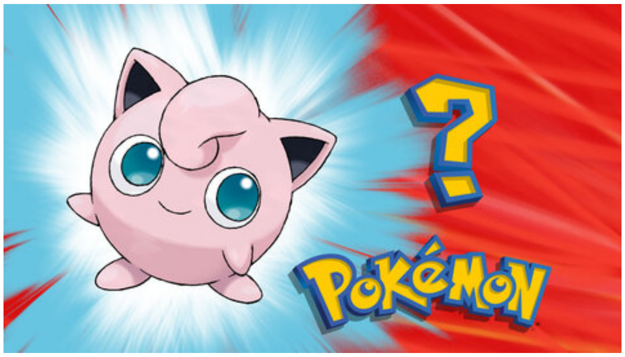
Knowledge Representation and Reasoning

Project Report

(Group 1)

Who's That Pokémon?



1. What does this project do?

The "Who's That Pokémon" project is an interactive game that challenges players to think of a specific Pokémon and then uses a knowledge-representation-based approach to guess its identity through a series of yes/no questions.

The project utilizes a rule-based system that populates a knowledge base with rules about Pokémon and uses forward-chaining to infer new rules and facts from existing knowledge. The user interacts with the system by answering yes/no questions asked by the system. Based on the answers given, the system updates its knowledge base and reasons further to narrow down the potential Pokémon that the user is thinking of.

The project is designed to be entertaining and engaging for players of all ages and levels of Pokémon expertise, while also demonstrating the capabilities of knowledge-based systems and reasoning techniques.

2. Why should we care about this project?

This project is inspired by the popular "Who's that Pokémon?" segment that used to air during commercial breaks in the Pokémon anime.

Our project provides a fun and engaging way for fans of the Pokémon franchise to test their knowledge and challenge themselves. It offers a unique challenge in the form of the AI's ability to reason and make deductions based on the information provided by the player.

Secondly, the game also serves as a tool for introducing new fans to the world of Pokémon. It allows them to learn about different Pokémon and their characteristics in a more interactive way, which can be a fun and memorable experience.

Finally, the project is a good example of the application of knowledge representation and reasoning. It combines forward-chaining and rule-based reasoning to make informed guesses and narrow down the potential Pokémon based on the user's responses to the questions asked. Overall, "Who's That Pokémon" is a project that is both entertaining and educational, appealing to both Pokémon fans and those interested in AI and knowledge representation.

3. Knowledge Representation

The knowledge representation in this project is based on inference rules, where statements about the Pokémon are represented as predicates and relations, such as "isLegendary" or "isColor Yellow". The system then uses these forward chaining to make inferences about the Pokémon that the user is thinking of.

In the context of our project, each rule in the knowledge base was represented as a Horn clause. For example, the rule:

((isColor Yellow) (canEvolve) … (isType Electric)) -> (isPokemon Pikachu)

where every part separated by parentheses represents logical conjunction (i.e., "and") and the → symbol represents logical implication (i.e., "if-then").

We used a text file to store the Horn clauses that represented our knowledge base. Each line in the file represented a single rule in the form of a Horn clause.

We also used permanent sub-rules in the form of Horn clauses to represent additional knowledge that could be used for inference. For example, the sub-rule:

(isColor Red) -> (isPrimaryColor)

Above represents the knowledge that all Pokémon that are colored red are also considered to be primary colored Pokemon. This allows us to ask broader questions and filter out more Pokémon on average.

By representing knowledge in the form of Horn clauses, we were able to use logical reasoning to infer new facts based on the user's input. This allowed us to narrow down the set of possible Pokémon that the user was thinking of, and eventually arrive at a single answer.

4. Reasoning

The system starts with zero facts in the knowledge base (KB) and starter rules of the form:

- rule: ((isColor Yellow) (isLegendary) … (isType Electric)) -> (isPokemon Zapdos)
- rule: ((isColor Yellow) (canEvolve) … (isType Electric)) -> (isPokemon Pikachu)
- rule: ((isColor Blue) (canEvolve) … (isType Water)) -> (isPokemon Squirtle)

Based on the user's answer to each question, we added new facts to the KB and derived new rules using the forward-chaining. We removed previous rules in the KB except for rules that could be used for inference.

After adding each fact, ask (isPokemon ?x) - the bindings returned for ?x contain our answer. Keep checking if there is only 1 rule with consequent (isPokemon ?x) in the KB at any point - that means we’ve narrowed it down to 1 Pokémon already and don’t need to ask any more questions.

We also have permanent sub-rules like:

rule: ((isColor Red)) -> (isPrimaryColor)

rule: ((isColor Yellow)) -> (isPrimaryColor)

rule: ((isColor Blue)) -> (isPrimaryColor) that are used for multilevel inference.

Iterate over all remaining Pokémon rules in the KB and pick the statement that best splits the solution space. Ask the user a question based on the selected statement to narrow down the possible Pokemon.

Example inference:

- After receiving a positive response from the user regarding the color yellow, add the fact "(isColor Yellow)" to the knowledge base.
- Derive:
  - rule: ((isLegendary) (isColor Yellow) … (isType Electric)) -> (isPokemon Zapdos)
  - rule: ((canEvolve) (isColor Yellow) … (isType Electric)) -> (isPokemon Pikachu)
- Remove all non-permanent rules previously in the KB.
- Add the new rules to the KB.

- Ask (isPokemon ?x) - the bindings returned for ?x contain our answer. Keep checking if there is only 1 rule with consequent (isPokemon ?x) in the KB at any point - that means we’ve narrowed it down to 1 Pokémon already and don’t need to ask any more questions.
- Iterate over all remaining Pokémon rules in the KB and pick the statement that best splits the solution space. Let’s say we picked the statement (isColor Red). Ask “Is the Pokémon you’re thinking of Red/Orange in color?” or something similar.

5. Running the Game

'Who's that Pokémon' system is designed to accurately guess the Pokémon that you have in mind by asking a series of questions and the answers it receives. With our user-friendly interface, you won't need to type anything, making the experience enjoyable for everyone regardless of their familiarity with Knowledge Representation and Reasoning.

To run the demo, please follow the instructions in the README.md file.