Solving Wordle with CSP

A constraint programming with LLM approach

Agenda

What's on the menu ?

1. What is Wordle ?
2. Some keywords
3. Constraint Solver
4. Hybrid Solver
5. Possible Improvements
6. Conclusion

What is Wordle ?

What is wordle ?

- A popular **single player word guessing game**
- Created by the **New York Times**
- Simple rules :
  - **Gray** - letter not in word
  - **Yellow** - in word but wrong position
  - **Green** - in word and right position

✅ Perfect problem for **Constraint Satisfaction Problem** (CSP) and **LLM** application

Some Keywords

Keywords

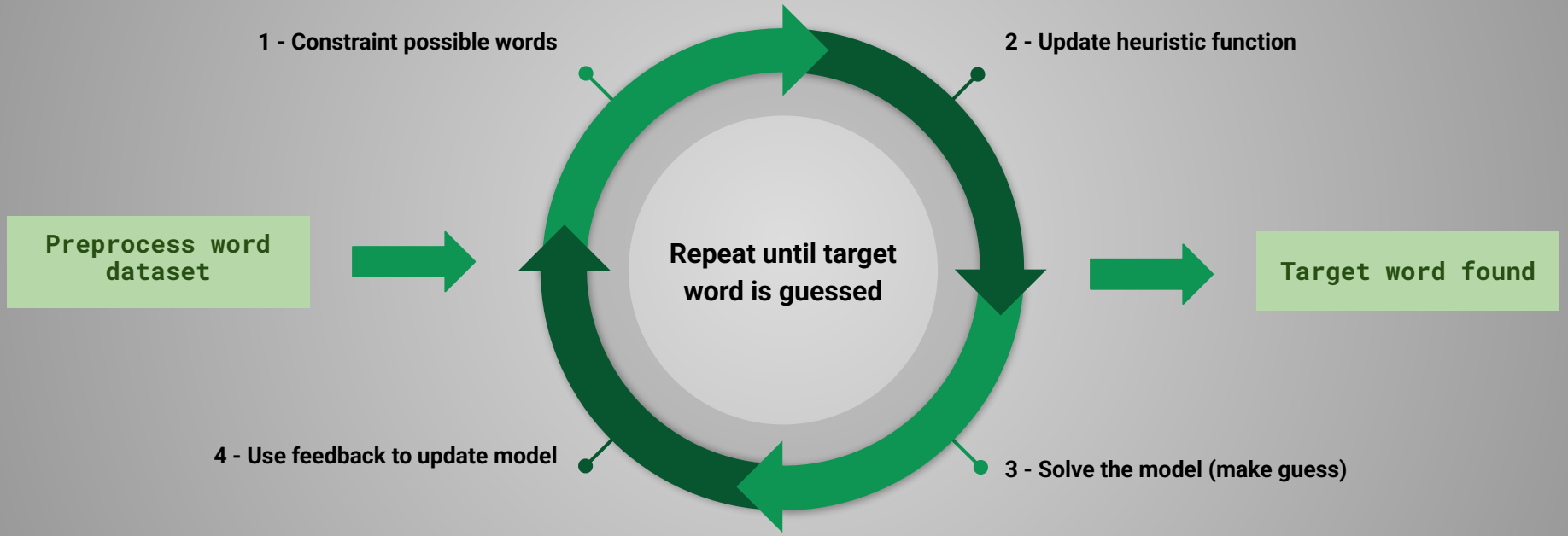
- **Constraint Satisfaction Problem (CSP):**
  - A computational problem defined by a set of variables, domains, and constraints that must all be satisfied simultaneously.

- **Heuristic (Computer science):**
  - A heuristic is an approach to problem-solving that employs a practical method not guaranteed to be optimal or perfect, but sufficient for reaching an immediate, short-term goal or approximate solution.

- **Large Language Model (LLM):**
  - A Large Language Model is a deep learning model, typically based on a transformer architecture, that uses billions (or even trillions) of parameters to perform natural language processing tasks such as text generation, translation, summarization, and question answering.

Constraint Solver

Constraint Solver - Main process



Constraint Solver - Implementation

🛠️ Technical stack:

- **Language**: Python
- **CSP Solver**: Google OR-Tools
- **Data processing**: Pandas

📊 Dataset:

- English word dataset (~16000 5 letter words)

🔢 Model constraints:

- Letter constraints (Gray / Yellow / Green)

- ***Heuristic function (maximize):***
  - **Word score = Cp * P + Cl * L - Cd * D**

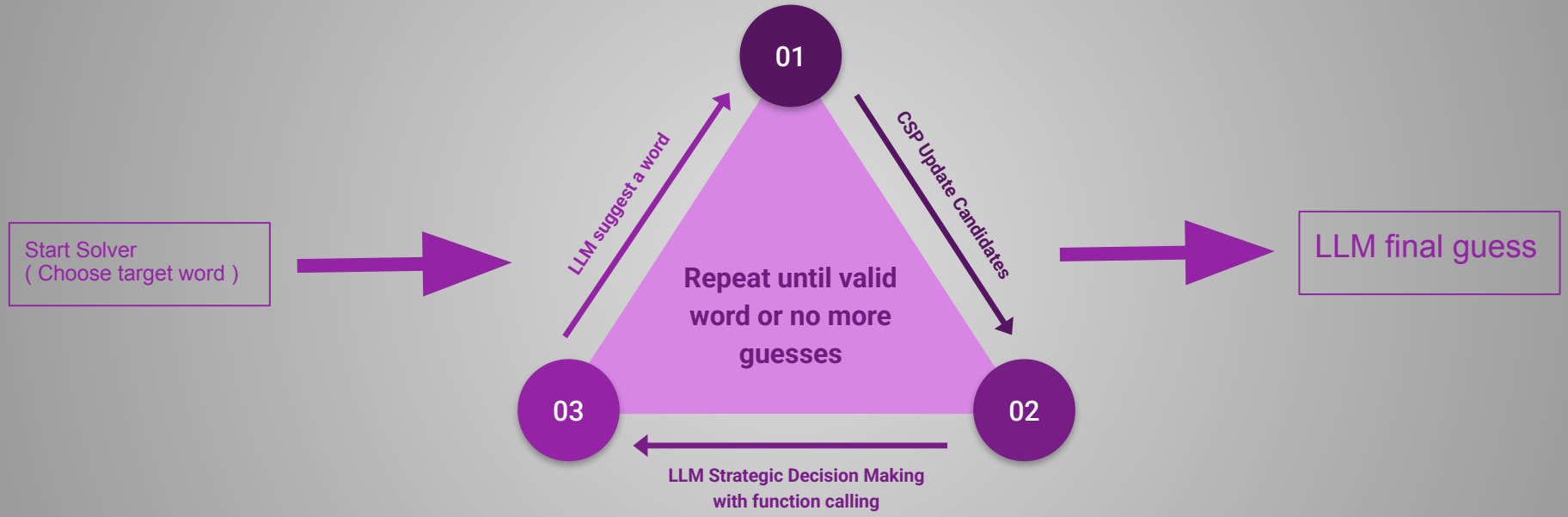
P = letter positional frequency
L = letter frequency
D = Nb duplicate letters in word
Cp = Coef. for P
Cl = Coef. for L
Cd = Coef. for D

Constraint Solver - Example output

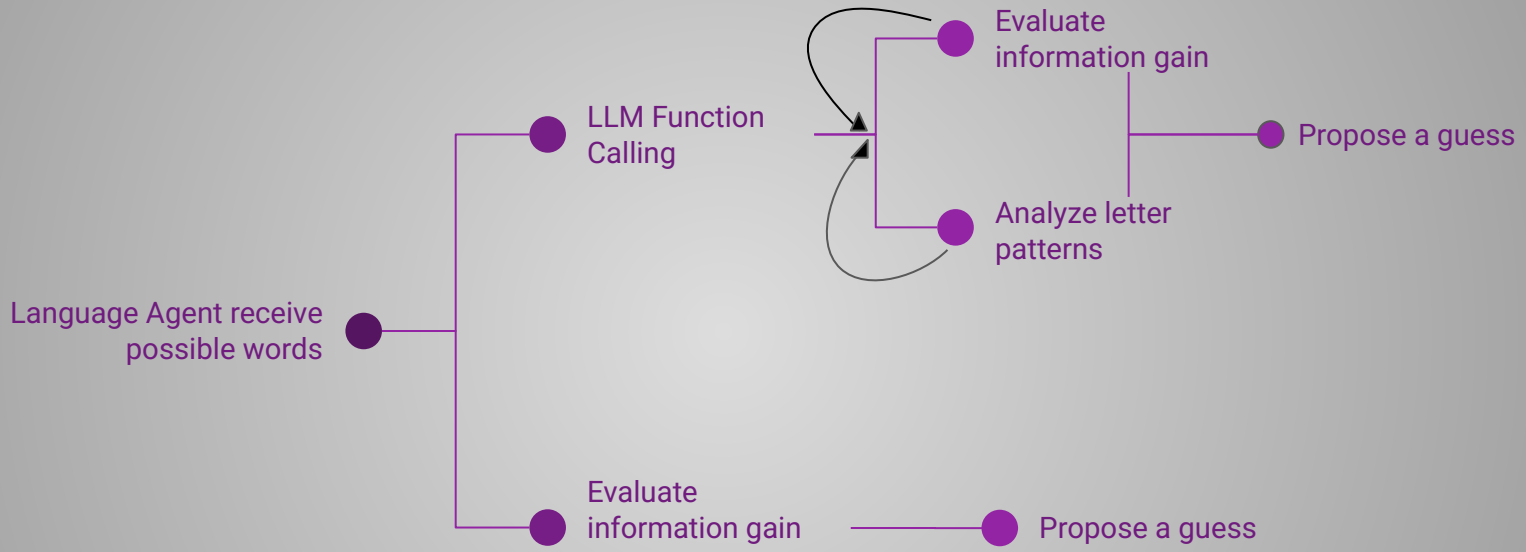
```
Random word: saury

Positional frequency: [defaultdict(<class 'int'>, {0: 0.073
{0: 0.5271653790591043, 4: 0.4901074053137366, 18: 0.410589
Length of possible words: 15921
Is saury in the dataset? True
status = OPTIMAL

Attempt 1: tares → ['B', 'G', 'Y', 'B', 'Y']
Length of possible words: 18
Is saury in the dataset? True
status = OPTIMAL

Attempt 2: sairy → ['G', 'G', 'B', 'G', 'G']
Length of possible words: 2
Is saury in the dataset? True
status = OPTIMAL

Attempt 3: saury → ['G', 'G', 'G', 'G', 'G']
Solved saury in 3 attempts!
```

Hybrid Solver

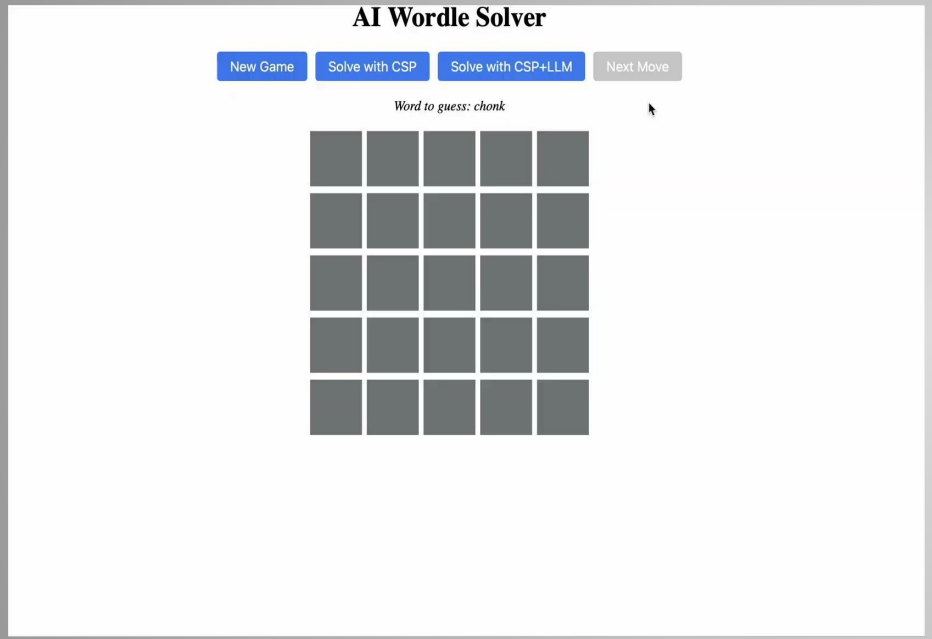
Hybrid-Solver - Solving Process

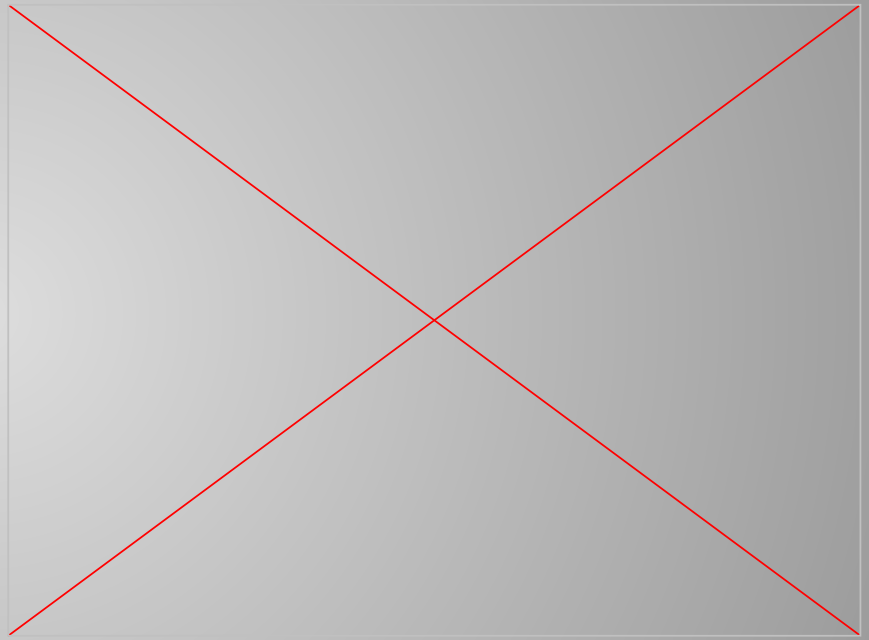


LLM Strategic Decision Making



Hybrid-Solver Demo





Possible Improvements

Possible improvements

CSP Solver

- Use bigger dataset
- Adversarial Filtering / Anti-Worst-Case
- Lookahead with Minimax Guessing

Hybrid-Solver

- Optimize Information Gain methods
- Enhanced LLM Prompting

Conclusion

Thank You!

Any questions ?

Contacts

leon.ayral@epita.fr
gabriel.calvente@epita.fr
cedric.damais@epita.fr
yacine.benihaddadene@epita.fr