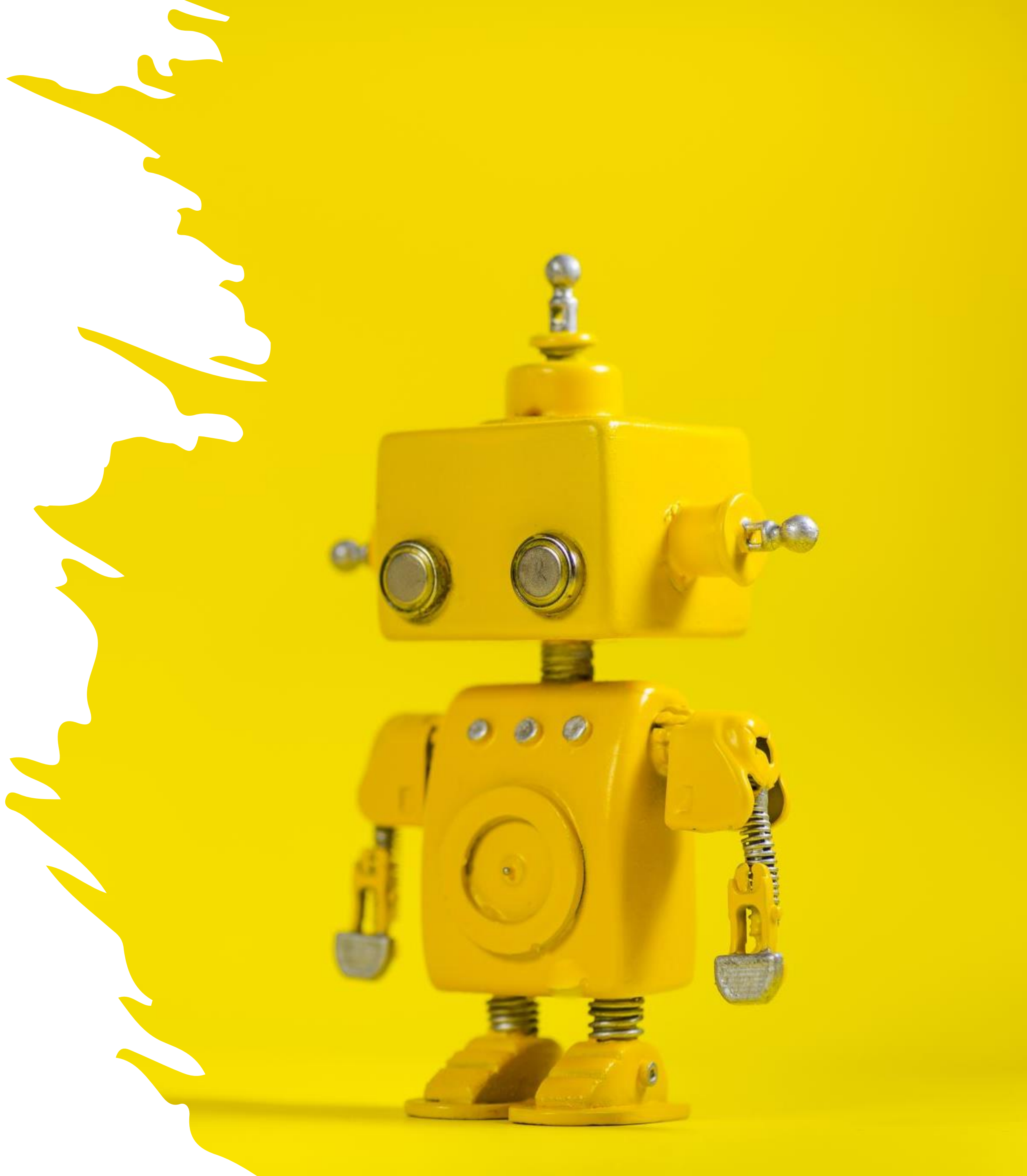
TidyBot: Personalized Robot Assistance with LLMs

Zoe Edgington and Liana Tutt



What is TidyBot?

- An LLM paired with an autonomous robot
- LLM is used to generalize user preferences about where they want common household items to be put away
  - Items like clothes, trash, aluminum cans
  - Put them in bins, furniture, drawers
  - The LLM takes these preferences and generates general preferences for the user
- The robot then cleans up the area by following these steps until no items remain
  - The perception system identifies the nearest item
  - The LLM uses the generalizes preferences to determine where to put it and what manipulation method to use to move the item
  - The robot picks up the item and moves it to the correct receptacle using the correct manipulation method

Utilizing LLM for Personalized Object Placement

- Purpose of LLM: To process user input about object placements and generate actionable rules that guide the robot's behavior in sorting items according to user preferences
- Operational Flow:
  - Users provide initial preferences in natural language, e.g., “yellow shirts go in the top drawer; navy socks in the bottom drawer.”
  - The LLM processes these preferences as Pythonic code, ensuring consistent input format for easy parsing and rule generation.
- LLM Output:
  - Generates a summary of rules from the processed inputs, such as “light-colored clothes in the top drawer, dark-colored clothes in the bottom drawer.”
- These summaries guide the robot in real-time to decide where new, unseen items should be placed based on learned preferences.

TidyBot’s LLM

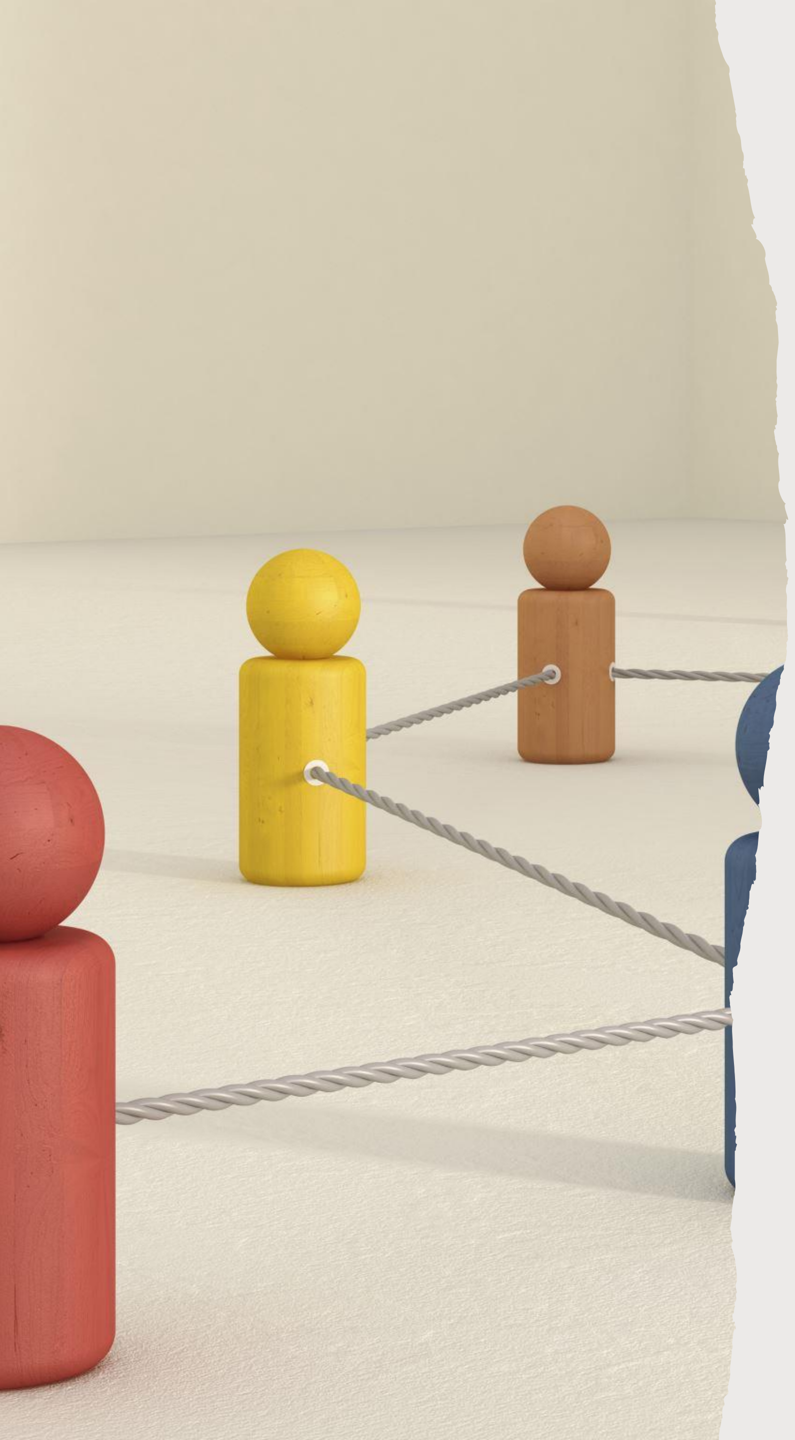
Visual of TidyBot taking user preferences and generalizing them [2]

```
objects = ["yellow shirt", "dark purple shirt",
"white socks", "black shirt"]
pick_and_place("yellow shirt")
pick_and_place("dark purple shirt")
pick_and_toss("white socks")
pick_and_place("black shirt")
# Summary: Pick and place shirts, pick and
toss socks.
```

TidyBot takes specific preferences and creates a summary of them [1]

```
# Summary: Pick and place shirts, pick and toss
socks.
objects = ["black socks", "white shirt", "navy
socks", "beige shirt"]
pick_and_toss("black socks")
pick_and_place("white shirt")
pick_and_toss("navy socks")
pick_and_place("beige shirt")
```

TidyBot predicts where to put unseen objects based on its summary [1]



Interaction Between Perception System and LLM

- Role of Perception System: Utilizes CLIP (Contrastive Language–Image Pre-training) to visually identify and localize objects within the environment.
- Integration with LLM:
    - The LLM’s output (summarized preferences) directs the CLIP model to focus on specific objects or categories relevant to current tasks, enhancing accuracy and relevance in object recognition.
    - Example: If the LLM determines that clothing items are of interest, the perception system prioritizes detection of clothes over other items.
- Dynamic Interaction:
    - Real-time feedback loop where the perception system updates the LLM with visual data, allowing for adjustments in task strategies based on current environmental context and object availability.

Demo

[2]

Benchmarking TidyBot's Performance

Benchmark Dataset Overview:

Comprises 96 scenarios across different room types (e.g., kitchen, bedroom) designed to test the LLM’s ability to generalize from seen to unseen object placements.

Each scenario includes a set of known (seen) placements and a challenge set of unknown (unseen) placements to evaluate generalization.

Testing Methodology:

The LLM’s predictions for unseen placements are compared against a ground truth to compute accuracy metrics, focusing on the robot’s ability to apply learned preferences to new situations.

Measures both the memorization of seen examples and the true generalization capability to unseen conditions.

Results:

91% accuracy on unseen objects in benchmark

85% accuracy on unseen objects in the real world

- Likely because of perception system

Contributions

- LLM text summarization provides generalization and user preferences in robotics
- A public benchmark for evaluating generalization of receptacle selection
- Implementation/evaluation of these concepts in real-world system

Extending the Work

- More personalized AI helpers
  - Can use the concept of generalized preferences from an LLM
  - Potential concepts
    - Robotic ingredient prepper
    - Personalized virtual assistants
- Put more LLMs into robots
  - Improve autonomous capabilities
    - Predicting needed action from object recognition and interpolation

Sources

- [1] Wu, J. et. al. (2023). TidyBot: Personalized robot assistance with large language models. *Autonomous Robots 47*(8), 1087–1102. https://doi.org/10.48550/arXiv.2305.05658
- [2] The Trustees of Princeton University. (2023, October 11). *Tidybot: Personalized Robot Assistance with large language models*. Princeton University. https://tidybot.cs.princeton.edu/