



Merle Pfau

I'm packing my  
suitcase.

LT2216 Dialogue Systems  
- Course Project



# The Game

- System and user take turns in adding objects to a suitcase, reciting all objects that have already been added



I'm packing a suitcase and I'm bringing toothpaste.

I'm packing a suitcase and I'm bringing toothpaste and tissues



That's right. I'm packing a suitcase and I'm bringing toothpaste, tissues, and a phone charger.

- You lose if you make a mistake in the reciting or try to add something that is already in the suitcase

# Technicalities

## RASA custom Actions

- ActionSetUpGame
- ActionExtractInput
- ActionPlay

## Error Handling

- FallbackPolicy: action\_restart  
utters: "Let's start over."

## Interactive Learning

```
class ActionExtractInput(Action):  
    def name(self):  
        return "action_extract_input"  
  
    def run(self, dispatcher, tracker, domain):  
  
        message = tracker.latest_message.get('text')  
        return [SlotSet('user_words', message)]
```

```
vocabulary = ["shoes",  
              "sunglasses",  
              "shirts",  
              "jackets",  
              "socks",  
              "pajamas",  
              "speakers",  
              "toothpaste",  
              ...]
```

# Challenges

- Highly dependant on Google's Speech Recognition
- Separating input into units
- User can make many unexpected mistakes



That's right. I'm packing a suitcase and I'm bringing toothpaste, tissues, and a phone charger.

I'm packing a suitcase and I'm bringing toothpaste issues a phone charger and a computer



You did not say tissues. You lost. Do you want to start over?

# Relation to Course Contents

- **Grounding**
  - used\_words = common ground
  - affirm + "I'm packing a suitcase and I'm bringing " + used\_words + word\_choice
- **Incrementality**
  - User and System incrementally add to common ground
  - User utterance processed incrementally
- **Spoken Dialogue** is different from written text
  - Tried to allow as much variance in the user input as possible

# Future Work

- **Limit user input**

- to actual words
- to the domain

→ Lower error rate

- Implement another way to make the **system lose**
  - e.g. 5% chance of making a mistake

# Demo

The screenshot displays the Google Actions Console interface for a project named 'PackingSuitcase'. The browser address bar shows the URL `console.actions.google.com/project/packingsuitcase/simulator/`. The console's top navigation bar includes tabs for Overview, Develop, Test (which is active), Deploy, and Analytics. On the right side of the console, there are icons for various actions and a user profile.

The main content area is divided into two panels. The left panel, titled 'Packing my suitcase', shows a simulated conversation. It begins with a system message: 'Hello! Welcome to the Game 'Packing my suitcase'. You can start by saying hi.' Below this, a user input 'hi' is shown. The system then responds with a detailed instruction: 'Hello dear opponent! Let's play a game. It's called I'm packing my suitcase. This is how it works: I will start by adding the first object: I'm packing a suitcase and I'm bringing shoes. You will then repeat this sentence and add another object to the suitcase, for example: I'm packing a suitcase and I'm bringing shoes and pants. Understood?'. At the bottom of this panel, there is a 'Suggested input' section with a 'cancel' button and a text input field containing the word 'yes'.

The right panel, titled 'Change version', displays configuration settings for the simulation. It includes dropdown menus for 'Surface' (set to 'Smart Displ...'), 'Language' (set to 'English (United States)'), and 'Location' (set to 'Uppsala, 753 20 Uppsala, Schweden'). Below these settings, there is a tabbed interface with 'DISPLAY' selected. The 'DISPLAY' tab shows a preview of the system's response to the user's input, which matches the text shown in the left panel. Other tabs include 'REQUEST', 'RESPONSE', 'AUDIO', 'DEBUG', and 'ERRORS'. A small video feed in the bottom right corner shows a person wearing headphones, likely the user or a demonstrator.

# Discussion

- Do you see any issues with how I read in the user input?  
(as a hard string, no entity extraction or similar)
- Is there a way the user could win or lose the game that I missed?

Thank you for your attention!