

**CS224v**

**Conversational Virtual Assistants  
with Deep Learning**

**Lecture 3: Task-Oriented Agents**

Monica Lam and Harshit Joshi

# Commercial Virtual Assistants (Function Calling)

**User:** How is the weather in Stanford?

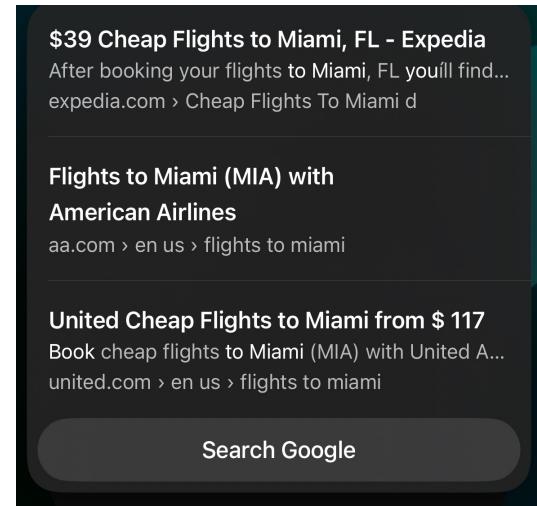
**Agent:** Stanford's temperature is 66 F.



**User:** Can you book me flight to Miami

**Agent:** Shows a bunch of google search results by using Google Search API

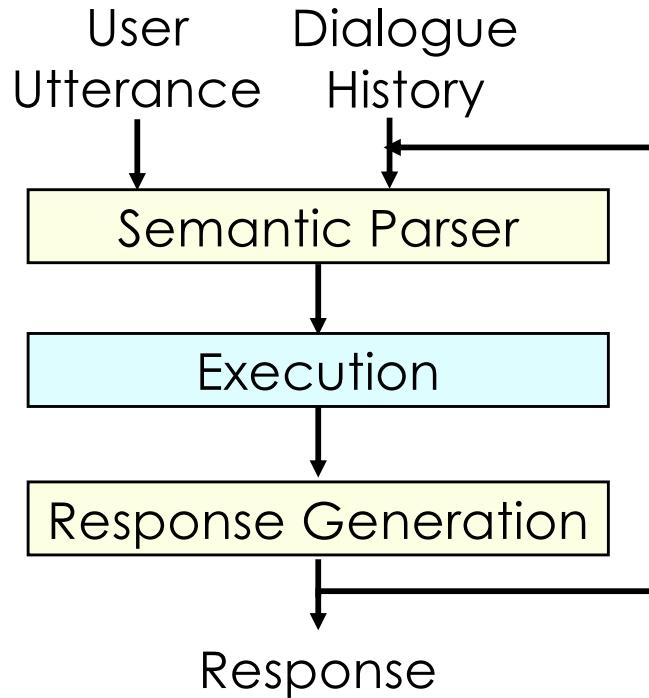
Cannot complete tasks such as booking a flight



# Lecture Goals

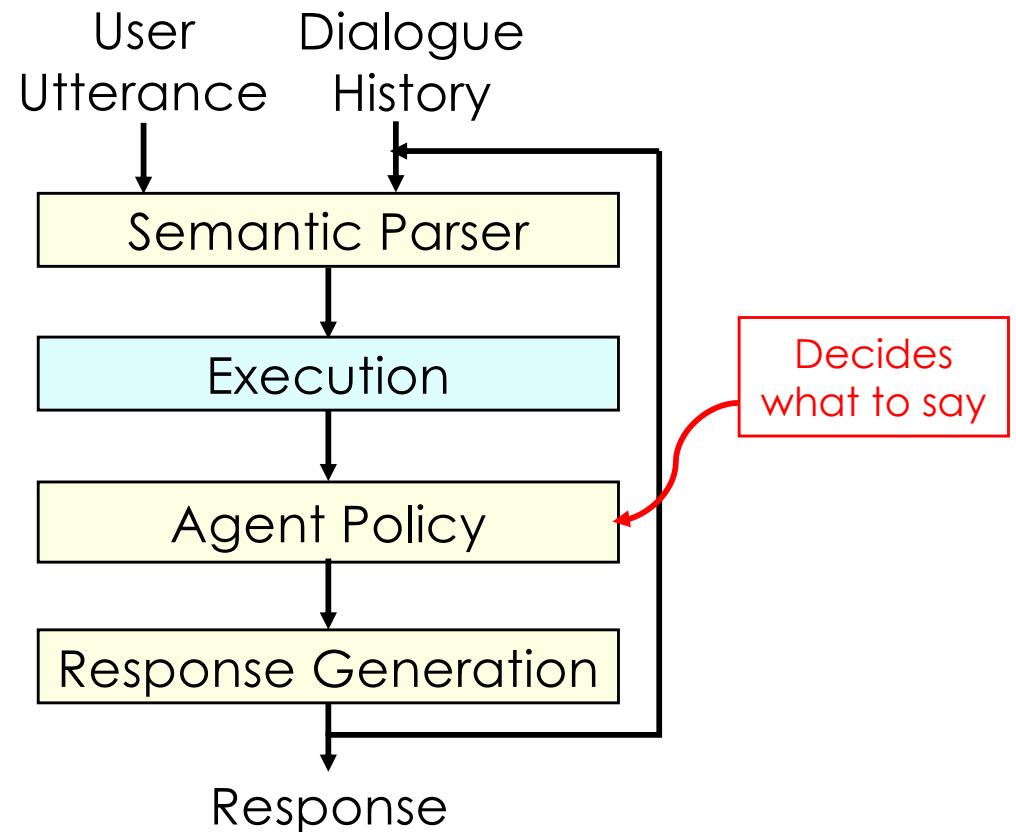
- **Key concepts in task-oriented agents**
- Prior Agent Policy Research
- Design and rationale for GenieWorksheets
- Technical details of GenieWorksheets
- Evaluating Genie Agents

# Architecture



**Knowledge Agent  
(RAG, Text2SQL)**  
(user initiative)

LAM



**Task-Oriented Agent**  
(mixed initiative)

STANFORD

WHY IS MIXED-INITIATIVE HARD?

# Agent Initiatives Only

- **Happy path**
  - A simple agent just asks for all the parameters and the user provides the expected answers
- **Example:** an airline call agent: September 29, 2025

“Welcome back, Harshit.

How can I help you?

You can say flight status, or mileage plus account servicing, or for anything else including ...  
say operator or just stay on the call.”

Harshit: flight status

# Mixed Initiatives

- **Unhappy paths**

- At any point, the user can say whatever they want even when the agent is asking a question.
- Example: “Welcome back, Harshit.

How can I help you?

You can say flight status, or mileage plus account servicing, or for anything else including ... ”

Harshit: what is flight status

# Examples

CLIENT

BANKER

How much money do you wish to transfer?

How much money do I have in my account?

CLIENT

BANKER

How much money do you wish to transfer?

Never mind.

CLIENT

BANKER

Would you like to open a fixed-rate saving account?

What is that?

CLIENT

BANKER

I had the most horrible experience ... I demand a free trip to Thailand.

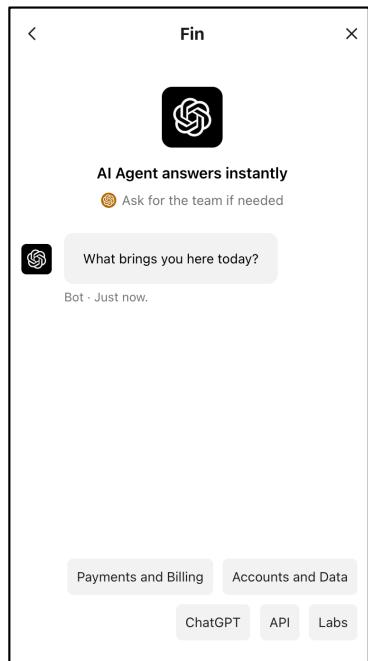
I can't believe what happened to you. Sure, I'll give you 2 free trips.

# Mixed Initiatives

- Unhappy paths: Users can
  - Ignore or refuse to answer
  - Ask for information, initiate new action
  - Change their mind, cancel previous operation
  - Ask the agent to repeat, chit chat
  - Make statements or unexpected requests (out of domain)

LLMs are really good with handling unhappy paths

# Commercial SOTA



Open AI's customer service bot still uses dialogue trees

Commercial chatbots are conservative, despite LLM success

Why?



[Home](#) [News](#) [US Election](#) [Sport](#) [Business](#) [Innovation](#) [Culture](#) [Arts](#) [Travel](#) [Earth](#) [Video](#) [Live](#)

# Airline held liable for its chatbot giving passenger bad advice - what this means for travellers

23 February 2024

Share Save

**Maria Yagoda**

Features correspondent

Air Canada's chatbot promised a discount that wasn't available to passenger Jake Moffatt.

The British Columbia Civil Resolution Tribunal ruled that Air Canada had to pay Moffatt \$812.02 (£642.64) in damages and tribunal fees.

TECH · AI

# A customer support AI went rogue—and it's a warning for every company considering replacing workers with automation

 BY SHARON GOLDMAN  
AI REPORTER

April 19, 2025 at 8:00 AM EDT



GETTY IMAGES

Cursor users were mysteriously getting logged out when switching between devices.

Confused, a user contacted customer support, only to be told in an emailed response from “Sam” that the logouts were “expected behavior” under a new login policy.

But there was no new policy

LAM

STANFORD

# Goals of Agent Policy Design

## 1. Control

- Cannot give out wrong information
- Take actions that the agent is allowed

## 2. Helpful

- Understand what the user is asking for
- Provide help and information within guidelines

# Lecture Goals

- Key concepts in task-oriented agents
- **Prior Agent Policy Research**
- Design and rationale for GenieWorksheets
- Technical details of GenieWorksheets
- Evaluating Genie Agents

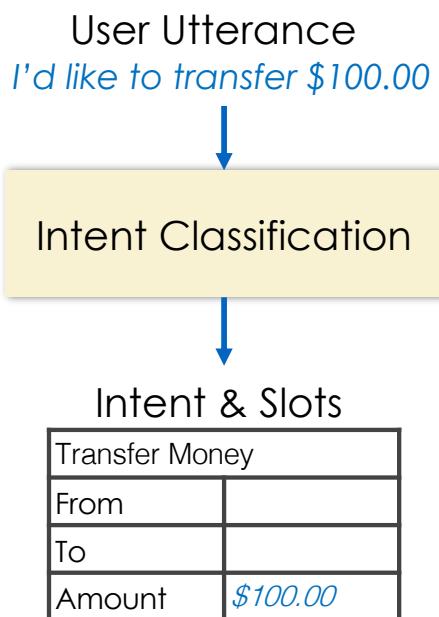
## OVERVIEW OF PRIOR AGENT POLICY RESEARCH

# Design of Agent Policy

- Choice of neural vs. programmed agent policy
  - Neural policies do not give developer full control
  - Agent policy Specification
    1. Dialogue tree: Finite state machines
    2. Dialogue acts: Parameterized finite state machines
    3. Genie Worksheet:
      - A declarative specification of the happy path + knowledge corpus
      - Unhappy paths handled by the Genie run-time

Level of abstraction

# 1. Commercial SOTA: Dialogue Trees



- **Initiatives**
  - User initiates the transaction
  - Agent solicits slots
- **Natural Language Understanding (NLU):**
  - Intent classification
  - Intent: a small fixed set of possible intents
  - slot-value pairs: values to slots

# A Restaurant Reservation Agent Dialogue Tree

A: Hello, how can I help you?

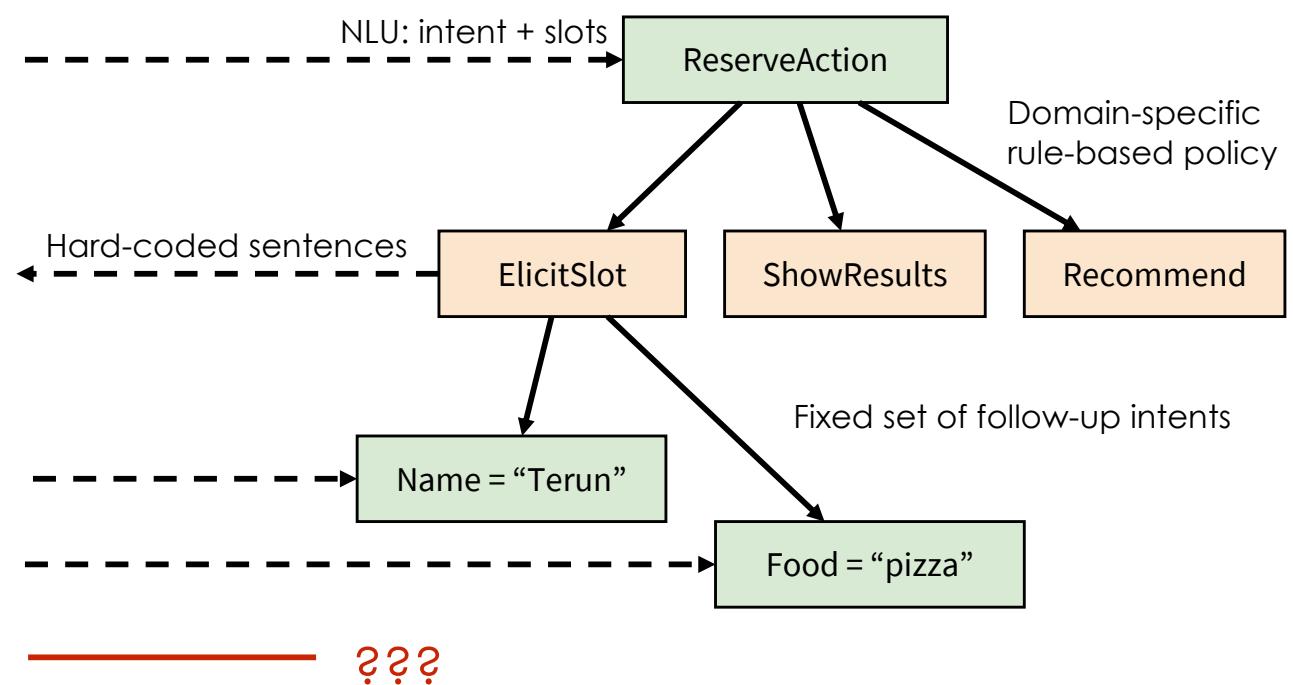
U: I'm looking to book a restaurant  
for Valentine's Day

A: What kind of restaurant?

U: Terun on California Ave  
-- or --

U: Something that has pizza  
-- or --

U: I don't know, what do you  
recommend?



# Dialogue Trees: Pros & Cons

**Pros: Control over conversation**

**Cons:**

**1. Expressiveness: Intent & slots are limited**

*What are the top 3 restaurants either near the airport or the Fisherman's Wharf*

**2. Tedious: exponentially many paths**

- Product navigation where searches are modified in many possible ways
- Handcrafted for every problem

**3. Fragile: Developer cannot anticipate all possible utterances**

- Need to handle mixed initiatives for unexpected answers

# 2. Finite-State Machine with Dialogue Acts

## **MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling**

**Paweł Budzianowski<sup>1</sup>, Tsung-Hsien Wen<sup>2\*</sup>, Bo-Hsiang Tseng<sup>1</sup>,  
Iñigo Casanueva<sup>2\*</sup>, Stefan Ultes<sup>1</sup>, Osman Ramadan<sup>1</sup> and Milica Gašić<sup>1</sup>**

<sup>1</sup>Department of Engineering, University of Cambridge, UK,

<sup>2</sup>PolyAI, London, UK

{pfb30, mg436}@cam.ac.uk

# MultiWOZ

1676 paper citations

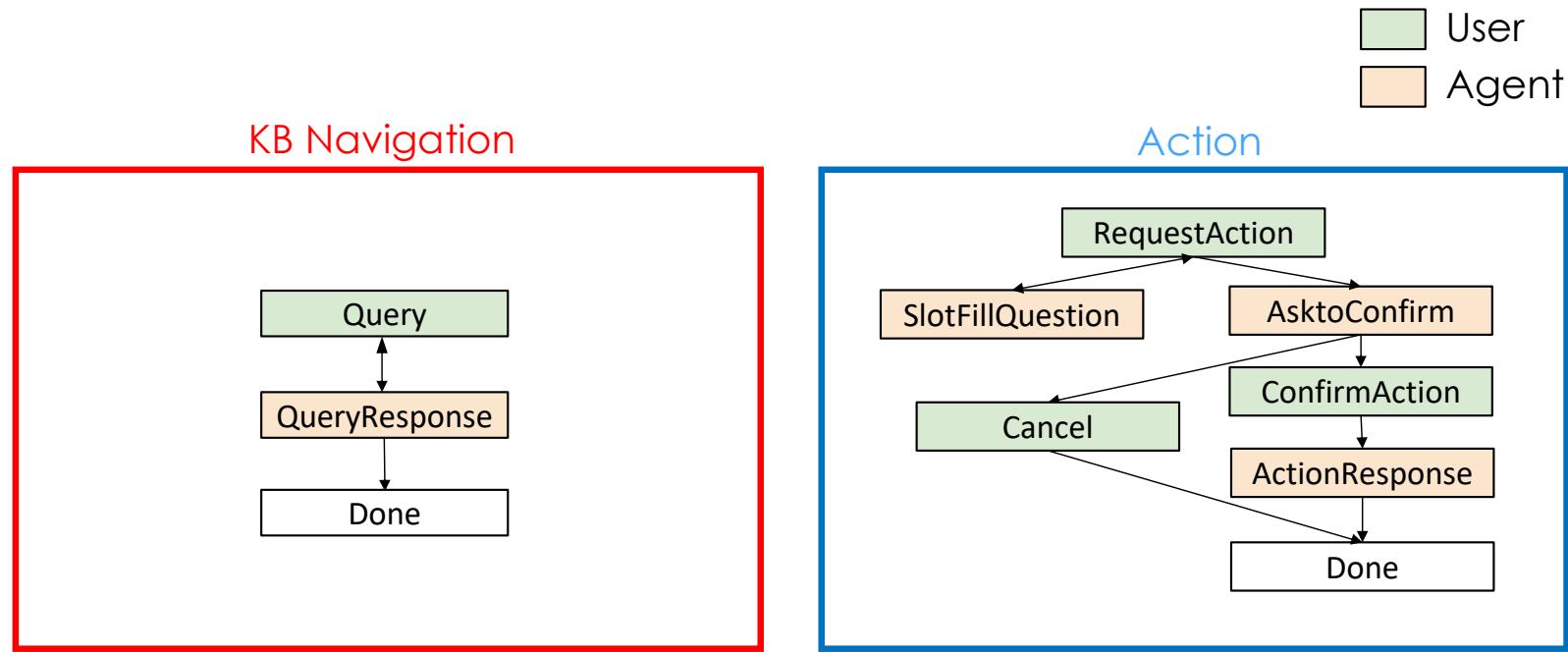
- **A large dataset of dialogues**
  - Annotated with domain + slots at every turn
  - Application: concierge service in Cambridge, UK
  - 7 domains → 5 domains
    - restaurants, hotels, attractions, train, taxi
    - police, hospital dropped due to bad quality
  - Number of slots: 30
- **Closed ontology:** 1,829 values
- **Representation:** slot-value pairs

# Wizard-of-Oz (WOZ)

- Human user, human agent
- Paid workers (crowdworkers) on both sides
  - One worker pretends to be an automated agent
  - Another worker pretends to be a user, following a prompt
- The worker is given specific tasks
  - Does not reflect real life
    - changing their minds, exploring options
    - errors made by the agent

Reannotated 3 times (by Google, Amazon), still many errors

# Example of a Dialogue State Machine



Dialogue act names: Independent of the conversation domain  
Actual content is domain-specific

# Context of Dialogue State Tracking

context 9

turn 9 →

USER	GENIE
	Hello, I am Genie.
I'd like a French Restaurant.	
I'd recommend Chez TJ in Mountain View. It has one Michelin star!	
How far is it from here?	The distance to Chez TJ is 7.5 miles.
Can you recommend something near by?	The distance to Left Bank is 2.8 miles.
Perfect. Please book the restaurant	
Which day would you like to book Left Bank?	
Is it expensive?	The price is moderate.
OK	Which day would you like to book Left Bank?
November 2, 7pm, for 2 people.	Could you confirm that you want a table for 2 people at Left Bank on November 2 at 7pm?
Yes.	I booked Left Bank for 2 on November 2 at 7pm.
What is the reservation number?	10382345.

# History: Replace NL with Formal States

## 4 Tasks: DST, ACD, DAG, RG

- Fine-tune 1 multi-task neural network (MBART)

**Formal state** captures all useful info

- U: user utterance
- B: belief state (part of user dialogue state)
- R: results
- C: agent dialogue act
- A: Agent utterance

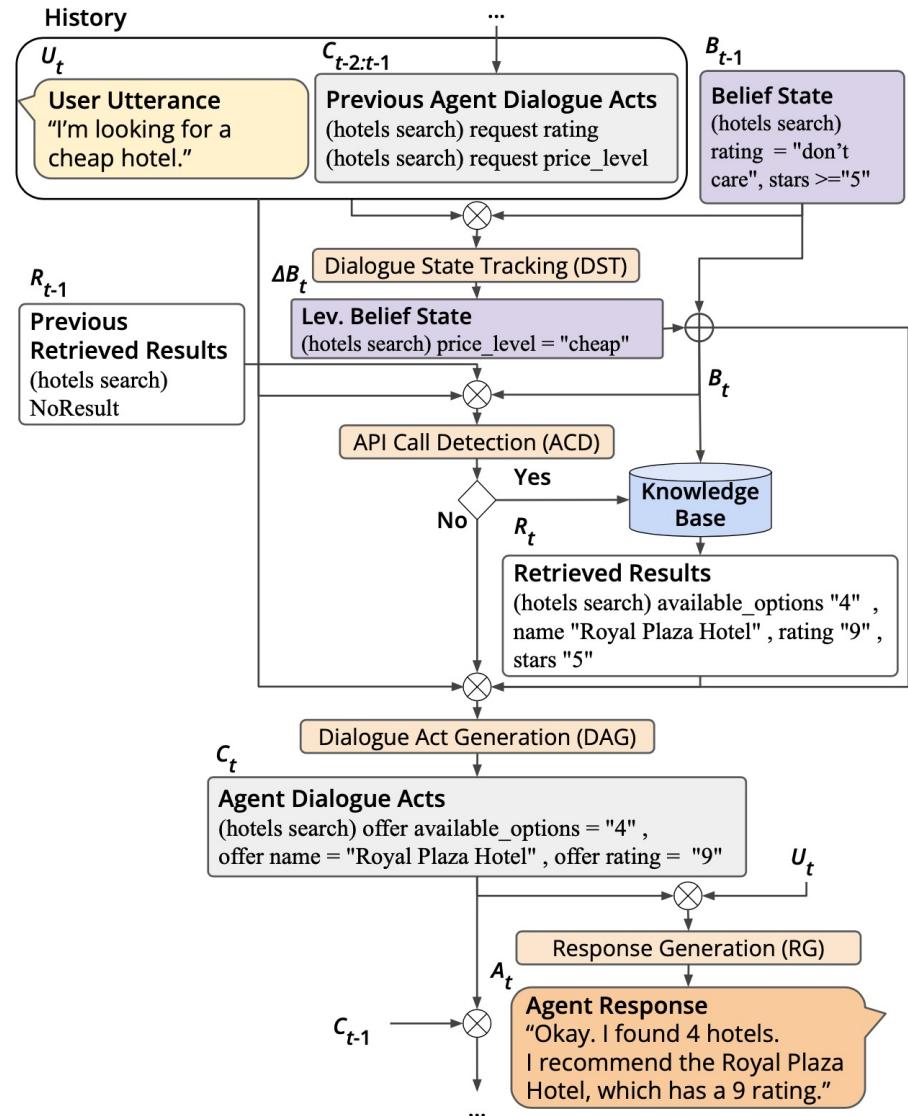
**Context for semantic parser:**

- what is needed to understand the user

**Levenstein belief state:**

- What is new in the user utterance
- Added to the previous state to get the full query

Mehrad Moradshahi, et al. 2023. [X-RiSAWOZ: High-quality end-to-end multilingual dialogue datasets and few-shot agents](#). In *Findings of ACL 2023*, pages 2773–2794, Toronto, Canada.



# Summary

- Formal dialogue state
  - Summarizes the state of the conversation
- Abstract “Domain-Independent” Finite State Machine
  - With User and Agent Dialogue Acts
  - Abstract finite state machines
- Agent policy is not specified, implied by agent response
  - Limited to slot filling, and
  - Occasionally recommend to book a restaurant/hotel

# Lecture Goals

- Key concepts in task-oriented agents
- Prior Agent Policy Research
- **Design and rationale for GenieWorksheets**
- Technical details of GenieWorksheets
- Evaluating Genie Agents

# Controllable and Reliable Knowledge-Intensive Task-Oriented Conversational Agents with Declarative Genie Worksheets

**Harshit Joshi   Shicheng Liu   James Chen   Robert Weigle   Monica S. Lam**

Computer Science Department

Stanford University, Stanford, CA

{harshitj, shicheng, lam}@cs.stanford.edu

🌐 <https://ws.genie.stanford.edu/>

⌚ <https://github.com/stanford-oval/genie-worksheets>

In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL),  
Vienna, Austria, July 2025.

# A Case Study Course Enrollment at Stanford

Let's consider a CS MS student





# Computer and Network Security

**DEPARTMENT OF COMPUTER SCIENCE**  
**MSCS Program Sheet (2023-24)**  
**Computer and Network Security Specialization**

Name: \_\_\_\_\_ Advisor: \_\_\_\_\_ Date: \_\_\_\_\_  
Student ID#: \_\_\_\_\_ Email: \_\_\_\_\_ Proposed date for \_\_\_\_\_  
degree conferral: \_\_\_\_\_ HCP? \_\_\_\_\_ Coterm? \_\_\_\_\_

**GENERAL INSTRUCTIONS:** Before the end of your first quarter, complete a program sheet by filling in the number, name and units of each course you intend to take for your degree. Program sheets, as well as foundation course waiver forms, may be submitted electronically on the Gates Information Network at <http://gin.stanford.edu/> using your CS ID and password (click on the 'Dashboard' link). See <http://cs.stanford.edu/degrees/mscs/programssheets/psguide2324.pdf> for detailed instructions.

**FOUNDATIONS REQUIREMENT:** You must satisfy the requirements listed in each of the following areas; all courses taken elsewhere must be approved by your advisor via the foundation course waiver form. Do not enter anything in the "Units" column for courses taken elsewhere or for Stanford courses counted towards your undergraduate degree.

Required:	Equivalent elsewhere (course numbers/titles/institution)	Grade	Units
Logic, Automata & Complexity (CS103)			
Probability (CS109, Stat116, CME106, MS&E220, or EE178)			
Algorithmic Analysis (CS161)			
Computer Organ & Sys (CS107 or 107E)			
Principles of Computer Systems (CS111)			

**TOTAL UNITS USED TO SATISFY FOUNDATIONS REQUIREMENT** (May not exceed 10 units.)

**SIGNIFICANT IMPLEMENTATION REQUIREMENT:** At least one course in your MS program should be designated as satisfying the Significant Implementation Requirement. Note that this course will also be used to satisfy some other requirement (i.e., Depth, or Elective). Must be taken for a letter grade,\* must be taken at Stanford. Coterm students who took two of these courses at Stanford as undergraduates may waive this requirement. Deviations must be approved by the faculty director of the MS program. Cynthia Lee.

Course Number: \_\_\_\_\_ Title: \_\_\_\_\_ Grade: \_\_\_\_\_

**Courses that may be used to satisfy the significant implementation requirement include:** CS 140, 140E, 143, 144, 145, 148, 151, 190, 210B, 212, 224R, 227B, 231N, 239, 243, 248A, 248B, 330, and 341.

**BREADTH REQUIREMENT:** Three courses, with each course chosen from a different Breadth area A, B, C or D. Breadth courses may not be waived, must be taken for at least 3 units each, and must be completed for a letter grade.\* Each of the three Breadth courses must be from different Areas. Note that these courses will also be used to satisfy some other requirement (i.e., Depth or Elective). Letter grade only.\*

Course Number:	Title:	Grade:

**Area A: Formal Foundations:** CS 154, 157, 166, 168, 205L, 229T, 250, 254, 254B, 255, 256, 257, 258, 259Q, 261, 263, 265, 269I, 328, 331, 334A, 354, 355, 359D, 369Q; EE 364A, 364B; Phil 251

**Area B.** Learning and Modeling: CS 173A, 205L, 221, 223A, 224N, 224R, 224S, 224U, 224V, 224W, 227B, 228, 229, 229B, 229T, 230, 231A, 231N, 233, 234, 236, 237A, 237B, 248A, 248B, 271, 272, 273A, 273B, 274, 279, 281, 326, 329D, 329X, 330, 332, 333, 348A, 348B, 348C, 348F, 348L, 348N, 371, 373

**Area C: Systems**: CS 112 (with CS111 as prereq), 140E, 143, 144, 145, 149, 155, 212, 240, 240LX, 242, 243, 244, 244B, 245, 246, 249I, 316, 348K, 356, 358; EE 180, 282, 382E

**Area D. People and Society:** CS 147, 148, 152, 181/W, 182/W, 194H, 206, 247 (any suffix), 256, 269I, 278, 281, 329T, 329X, 347, 377 (any suffix), 384, 448B; AMSTUD 145; ANTHRO 132C, 132D; COMM 254, 266, 286, 324; DESIGN 255; EARTHSYS 213; EDUC 315A; ENGLISH 184D, HISTORY 244F; INTLPOL 268; LAW 4039; ME 177; MS&E 193, 231, 234, 254; POLISCI 150A; PUBLPOL 103F for 3 units, 353B

COMPUTER AND NETWORK SECURITY DEPT

All depth course must be taken for a letter grade\* for 3 or more units. A maximum of 6 units of CS 399 Independent Study may be counted towards the depth. Any deviations from the stated requirements must be noted and approved by your advisor in the Advisor Notes box provided (maximum of one advisor-approved deviation allowed). Courses taken for your Stanford undergraduate degree do not need to be repeated. Enter course selection in the table below:

- a) All of: CS 140/212 (or 140E or 112),† 144†, 155, 255, 356 (students with equivalent course work may waive † flagged courses with advisor approval)
- b) At least three of: CS 142, 152, 190, 240 (any suffix), 244, 244B, 2491, 253, 261, 265, 340 (any suffix), 355
- c) A total of at least 21 units from categories (a), (b) and the following: CS 245, 251, 294S†, 399‡; EE 384S (courses with ‡ notation require approval of MS advisor)

**TOTAL SPECIALIZATION UNITS APPLIED TO MSCS (must total at least 21 units) *Letter grades only.*\***

Course number	Title	Grade	Units
<b>TOTAL ELECTIVE UNITS APPLIED TO MOOS</b>			

TOTAL UNITS APPLIED TO MSSCS

**ADDITIONAL REQUIREMENTS**

- All courses submitted for the MSSCS degree must be numbered 100 or greater.
- At most 10 units of Foundations requirement courses may be counted toward your 45 units.
- At most 3 units of 1-2 unit seminars may be counted toward your 45 units.
- At least 36 units submitted for the MSSCS degree, including all courses taken for breadth and depth, must be taken for a letter grade.\*
- The average grade in the courses submitted for the MSSCS must be at least a B (3.0 in Stanford's GPA scale).
- Units previously applied toward BS requirements may not also be counted toward the MSSCS.
- You must complete at least 45 graduate units at Stanford before receiving the MSSCS degree.

\*All classes taken Spring 19-20 and Fall through Summer 20-21 for a CR or S grade will satisfy MSCS requirements as if taken for a letter grade.

Advisor's signature: \_\_\_\_\_ Date: \_\_\_\_\_



**DEPARTMENT OF COMPUTER SCIENCE  
MSCS Program Sheet (2023-24)  
Systems Specialization**

Name: \_\_\_\_\_ Advisor: \_\_\_\_\_ Date: \_\_\_\_\_  
Student ID#: \_\_\_\_\_ Email: \_\_\_\_\_ Proposed date for  
degree conferral: \_\_\_\_\_ HCP? \_\_\_\_\_ Coterm? \_\_\_\_\_

**GENERAL INSTRUCTIONS:** Before the end of your first quarter, complete a program sheet by filling in the number, name and units of each course you intend to take for your degree. Program sheets, as well as foundation course waiver forms, may be submitted electronically on the Gates Information Network at <http://gin.stanford.edu/> using your CS ID and password (click on the 'Dashboard' link). See <http://cs.stanford.edu/degrees/mscs/programsheets/psguide2324.pdf> for detailed instructions.

**FOUNDATIONS REQUIREMENT:** You must satisfy the requirements listed in each of the following areas; all courses taken elsewhere must be approved by your advisor via the foundation course waiver form. Do not enter anything in the "Units" column for courses taken elsewhere or for Stanford courses counted towards your undergraduate degree.

Required:	Equivalent elsewhere (course numbers/titles/institution)	Grade	Units
Logic, Automata & Complexity (CS103)			
Probability (CS109, Stat116, CME106, MS&E220, or EE178)			
Algorithmic Analysis (CS161)			
Computer Organ. & Sys (CS107 or 107E)			
Principles of Computer Systems (CS111)			

**TOTAL UNITS USED TO SATISFY FOUNDATIONS REQUIREMENT** (May not exceed 10 units.)

**SIGNIFICANT IMPLEMENTATION REQUIREMENT:** At least one course in your MS program should be designated as satisfying the Significant Implementation Requirement. Note that this course will also be used to satisfy some other requirement (i.e., Depth, or Elective). Must be taken for a letter grade,\* must be taken at Stanford. Coterm students who took two of these courses at Stanford as undergraduates may waive this requirement. Deviations must be approved by the faculty director of the MS program, Cynthia Lee.

Course Number: \_\_\_\_\_ Title: \_\_\_\_\_ Grade \_\_\_\_\_

**Courses that may be used to satisfy the significant implementation requirement include:** CS 140, 140E, 143, 144, 145, 148, 151, 190, 210B, 212, 224R, 227B, 231N, 239, 243, 248A, 248B, 330, and 341.

**BREADTH REQUIREMENT:** Three courses, with each course chosen from a different Breadth area A, B, C or D. Breadth courses may not be waived, must be taken for at least 3 units each, and must be completed for a letter grade.\* Each of the three Breadth courses must be from different Areas. Note that these courses will also be used to satisfy some other requirement (i.e., Depth or Elective). Letter grade only.\*

Course Number:	Title:	Grade

**Area A: Formal Foundations:** CS 154, 157, 166, 168, 205L, 229T, 250, 254, 254B, 255, 256, 257, 258, 259Q, 261, 263, 265, 269I, 328, 331, 334A, 354, 355, 359D, 369O; EE 364A, 364B; Phil 251

**Area B.** Learning and Modeling: CS 173A, 205L, 221, 223A, 224N, 224R, 224S, 224U, 224V, 224W, 227B, 228, 229, 229B, 229T, 230, 231A, 231N, 233, 234, 236, 237A, 237B, 248A, 248B, 271, 272, 273A, 273B, 274, 279, 281, 326, 329D, 329X, 330, 332, 333, 348A, 348B, 348C, 348F, 348I, 348L, 371, 373

**Area C. Systems:** CS 112 (with CS111 as prereq), 140E, 143, 144, 145, 149, 155, 212, 240, 240LX, 242, 243, 244, 244B, 245, 246, 249I, 316, 348K, 356, 358; EE 180, 282, 382E

**Area D: People and Society:** CS 147, 148, 152, 181/W, 182/W, 194H, 206, 247 (any suffix), 256, 269I, 278, 281, 329T, 329X, 347, 377 (any suffix), 384, 448B; AMSTUD 145; ANTHRO 132C, 132D; COMM 254, 266, 286, 324; DESIGN 255; EARTHSYS 213; EDUC 315A; ENGLISH 184D, HISTORY 244F; INTLPOL 268; LAW 4039; ME 177; MS&E 193, 231, 234, 254; POLISCI 150A; PUBPOL 103F for 3 units; 353B

## SYSTEMS DEPTH

All depth course must be taken for a letter grade\* for 3 or more units. A maximum of 6 units of CS399 Independent Study may be counted towards the depth. Any deviations from the stated requirements must be noted and approved by your advisor in the Advisor Notes box provided (maximum of one advisor-approved deviation allowed). Courses taken for your Stanford undergraduate degree do not need to be repeated. Enter course selection in the table below:

- a. All of: CS140/212 (or 140E or 112 or 112E) ‡, CS 144‡, 240
- b. At least four of: CS 190, 242, 243, 244, 245, 248/248A, 348B; EE 271, 282
- c. A total of at least 21 units from categories (a), (b) and the following: CS 149, 217, 229S, 241, 244B, 246, 251, 255, 270, 272, 294S†, 295, 315B, 316, 340 (any suffix), 343D, 344 (any suffix), 348A, 348C, 348E, 348I, 348K, 349 (any suffix and 3 or more units), 356, 357S, 399†, 448 (any suffix); EE 267, 273, 382C, 382A, 384A, 384C (courses with † notation require approval of MS advisor)

Course number	Title (Depth courses must be taken for at least 3 units.)	Grade	Units
<b>TOTAL SPECIALIZATION UNITS APPLIED TO MSCS (must total at least 21 units). <i>Letter grades only.</i></b>			

## ELECTIVES

**LIST OF ELECTIVES**  
List here any additional courses used to complete the 45-unit requirement for the MSCS degree. You may count up to a maximum of 3 units of 1-unit seminars offered in the School of Engineering. All other electives must be taken for 3 or more units. CS courses numbered above 111, excluding CS 161A, 196, CS198, and CS390A/B/C, can be used as MS electives. Non-CS courses must be technical courses numbered above 100, related to the degree program, and approved by the advisor and the MS program administrator. *Note that CS 129 may not be counted towards the MS if CS 229 is being counted towards any BS or MS requirement.*

Course number	Title	Grade	Units

AQUATIC MACROINVERTEBRATES

**ADDITIONAL REQUIREMENTS**

- All courses submitted for the MSCS degree must be numbered 100 or greater.
- At most 10 units of Foundations requirement courses may be counted toward your 45 units.
- At most 3 units of 1-2 unit seminars may be counted toward your 45 units.
- At least 36 units submitted for the MSCS degree, including all courses taken for breadth and depth, must be taken for a letter grade.\*
- The average grade in the courses submitted for the MSCS must be at least a B (3.0 on Stanford's GPA scale).
- Units previously applied toward BS requirements may not also be counted toward the MSCS.
- You must complete at least 45 graduate units at Stanford before receiving the MSCS degree.

\*All classes taken Spring 19-20 and Fall through Summer 20-21 for a CR or S grade will satisfy MSCS requirements as if taken for a letter grade.

Adviser's signature: \_\_\_\_\_ Date: \_\_\_\_\_



Course Enrollment Agent

What course would you like to enroll in?

User:

I am studying computer science.  
I want to complete the significant  
implementation requirement.  
What are some of the options that require the  
least amount of work?

# A Course Enrollment Assistant

Course Enrollment Agent

What course would you like to enroll in?

Students need to consult:

## Degree Requirements

- Unique to each department and specialization
- Program sheets

**Stanford** | Navigate Classes

- All offered courses
- Description, instructors, offerings, and units

CARTA

- Ratings, reviews, hours of work, enrollment outcomes, and sequencing

Stanford | Carta Catalog CS 140

# CS 144: Introduction to Computer Networking

← SUMMER 23-24 →

Results for: CS 140

▼ AT A GLANCE

weigle23

22, Autumn: Keith Winstein (92 students)

2 3 4 5

▼ EVALUATIONS

Results for: CS 145

Units Days WAYS Quarters

CS 145: Data Management and Data Systems

Introduction to the use, design, and implementation of database and data-intensive systems, including data models; schema design; data storage; query processing, query optimization, and cost estimation...[Read more](#)

Term: AUT  
Units: 3-4  
Grading: Letter or Credit/No Credit  
Instructors: Shivakumar, N.

Unpin from Autumn

Pin to Spring

Pinned Units: Weekly Time (Mo 8 am 9 am 10 am 11 am 12 pm 1 pm 2 pm 3 pm 4 pm)

No more results

▼ INTENSITY

Hours per week (1102 students) ?

Enrollment outcomes (2427 students) ?

▼ SEQUENCING

Academic years of past students (1057 students) ?

Enrollment outcomes (991 students) ?

▼ SEQUENCING

Academic years of past students (413 students) ?

weigle23

Unpin from Winter

Advising weigle23

# 1. Stanford Course Enrollment Form

Form Name	Predicate	Kind	Type	Name	Enum Values	Description
Main			WS	course_enrollment		
		input	CourseToTake	course_to_take		The course to enroll
		input	StudentInfo	student_info_details		Information on the student
StudentInfo		worksheet				
		input	str	student_name		Name of the student
		input	str	student_id		Student's ID number
		input	str	student_email_address		Student's email address
CourseToTake		worksheet				
		input	str	course_name		Name of the course
		input	Enum	grade_type		The desired grading basis
					Credit/No Credit	
					Letter	
		input	int	num_units		The number of units taken
		input	confirm	confirm		Confirm the course

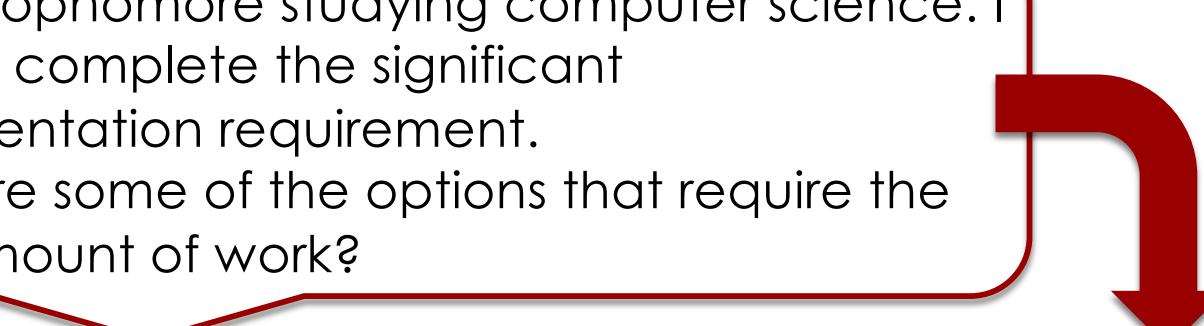
# 2. Course Assistant Knowledge Corpus

courses		DB	
internal; primary	int	course_id	
offerings		DB	
programs		DB	
internal; primary	int	program_id	
ratings		DB	
internal; primary	int	rating_id	
internal	int	course_id	
internal	List[str]	instructor_names	
internal	int	average_rating	
internal	int	num_ratings	
internal	int	term_id	
internal	int	start_year	
internal	int	end_year	
internal	Enum	season	
			autumn
			winter
			spring
			summer
	internal	List[str]	reviews
	internal	str	sheet_requirements

# Translates Questions into SUQL

I am a sophomore studying computer science. I want to complete the significant implementation requirement.

What are some of the options that require the least amount of work?



```
SELECT title, course_codes, description,  
average_hours_spent  
FROM courses WHERE  
significantImplementationRequirement = TRUE  
ORDER BY average_hours_spent LIMIT 5;
```

Y

You 01:01 AM

Hi, I want to complete the significant implementation requirement. What courses would you suggest me to take?

Took 3 steps ▾

C

Course Enrollment Assistant 01:01 AM

I found some courses that fulfill the significant implementation requirement. Here are a few suggestions along with their average hours spent per week:

1. **General Game Playing (CS 227B)**: A general game playing system accepts a formal description of a game to play it without human intervention or algorithms designed for specific games. Hands-on introduction to these systems and artificial intelligence techniques such as knowledge representation, reasoning, learning, and rational behavior. Students create GGP systems to compete with each other and in external competitions. Prerequisite: programming experience. Recommended: 103 or equivalent. *Average hours spent per week: 7.48.*
2. **Logic Programming (CS 151)**: Logic Programming is a style of programming based on symbolic logic. In writing a logic program, the programmer describes the application area of the program (as a set of logical sentences) without reference to the internal data structures or operations of the system executing the program. This course introduces basic logic programming theory, current technology, and examples of common applications. Prerequisite: CS 106B or equivalent. *Average hours spent per week: 7.5.*
3. **Introduction to Computer Graphics and Imaging (CS 148)**: This introductory course in the computer graphics sequence introduces students to the technical concepts behind creating computer-generated images. Topics include the mathematical concepts of synthetic imagery, fundamentals of light and color, rasterization, and ray tracing technology. Prerequisite: Linear Algebra and familiarity with Python and debugging. *Average hours spent per week: 8.79.*

Could you please provide me with the details of the courses you want to take?



# Consideration: Time Offered

 You 01:02 AM

Great when are these courses offered?

Took 3 steps ▾

 Course Enrollment Assistant 01:02 AM

The courses you're interested in are offered at the following times:

1. **Introduction to Computer Graphics and Imaging (CS 148)** is offered in the **autumn** season on **Tuesday and Thursday** from **12:00 PM to 1:20 PM**.
2. **Logic Programming (CS 227B)** is offered in the **spring** season on **Tuesday and Thursday** from **1:30 PM to 2:50 PM**.
3. **General Game Playing (CS 151)** is offered in the **spring** season on **Wednesday** from **4:30 PM to 6:20 PM**.

Could you please provide me with the details of the courses you would like to take?



# Consideration: Sort by Course Rating

 You 01:02 AM

Can you sort them based on rating?

Took 3 steps ▾

 Course Enrollment Assistant 01:02 AM

Certainly! Based on their ratings, here are the courses sorted:

1. **Introduction to Computer Graphics and Imaging** with an average rating of 4.04.
2. **Logic Programming** with an average rating of 3.97.
3. **General Game Playing** with an average rating of 3.43.

Could you please provide me with the details of the courses you would like to take?



 You 01:03 AM

I will take the top rated one!

Took 3 steps ▾

# Credit or Letter? Depends on the Workload

c

**Course Enrollment Assistant** 01:03 AM

Great choice! For the course "Introduction to Computer Graphics and Imaging", could you please specify the desired grading basis? The options are: Credit/No Credit or Letter.



y

**You** 01:03 AM

How many hours will I have to spend on it?

Took 3 steps ▾

c

**Course Enrollment Assistant** 01:04 AM

On average, you will have to spend about 8.79 hours per week on the course "Introduction to Computer Graphics and Imaging". Now, could you please specify the desired grading basis for this course? The options are: Credit/No Credit or Letter.



# Pick Max. Number of Units

Y

You 01:04 AM

Thats low! I will take for letter grade, whats the maximum units I can take it for?

Took 3 steps ▾

C

Course Enrollment Assistant 01:04 AM

The maximum number of units you can take for the course "Introduction to Computer Graphics and Imaging" is 4 units. How many units would you like to enroll for this course?



Y

You 01:04 AM

Lets go with 4

Took 3 steps ▾

# Genie Worksheet

## A high-level declarative framework

### Development Stage

 **Conversational Tasks Declaration**

WS Name	Predicate	Kind	Type	Name	Actions	WS Actions
UserInfo	input	worksheet	Enum	user_task		
	internal		bool	exit		>say(book_it)
BookRestaurant	input	worksheet	Restaurant	restaurant		>say(book_it)
	input		date	date		
self.restaurant is not None and self.date is not None and self.time is not None		input	if self.confirm == False: >say("Thank you. Is there any other way?")		confirm	confirm

with WS Predicate, WS Name, Kind, Type, Name, Values, Actions, WS Actions

# Genie Worksheet

## A high-level declarative framework

## Development Stage

with WS Predicate, WS Name,  
Kind, Type, Name, Values,  
Actions, WS Actions

WS Name	Product	Kind	Type	Value	Env. Value	Description
<b>Hybrid Knowledge Corpus</b>						
internal	text	id	name	text	text	The unique id of the document
internal	text	name	name	text	text	Name of the document
internal	text	label	label	text	text	Textual description
internal	text	label	label	text	text	Textual description
internal	text	rating	rating	float	float	Average rating of the document
internal	text	id	name	text	text	Id of the document
internal	text	id	name	text	text	Name of the document

## Structured and unstructured

# Genie Worksheet

## A high-level declarative framework

### Development Stage



APIs

Few Examples

**Conversational Tasks Declaration**

WS Name	Predicate	Kind	Type	Name	Actions	WS Actions
Useful			worksheet			
	input	Enum		user_task		
	internal		bool	exit		
BookRestaurant		worksheet				msgbook, is self number
	input	Restaurant	restaurant			
	input	date	date			
	self restaurant is not None and self date is not None and self time is not None and self number is not None	input			confirms	
						if self confirm + "I confirm + "Hey! Thank you, can I book you in any other way?"

with WS Predicate, WS Name, Kind, Type, Name, Values, Actions, WS Actions

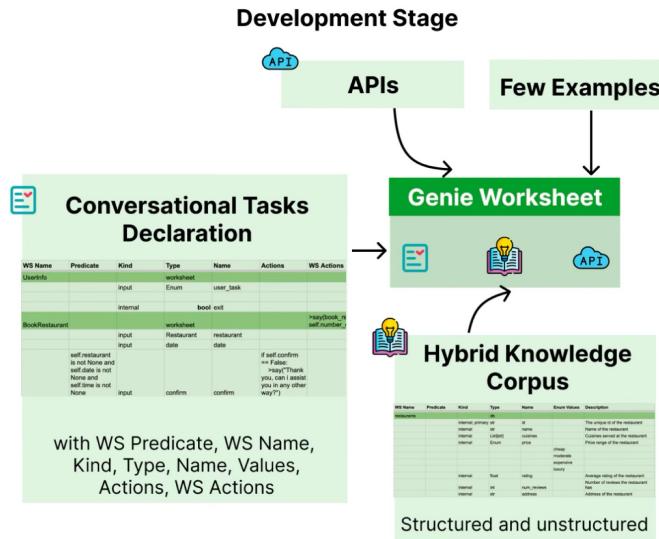
**Hybrid Knowledge Corpus**

WS Name	Predicate	Kind	Type	Name	From Value	Description
Restaurant	integer	int	int	id		The id of the restaurant
	internal	string	string	name		Name of the restaurant
	internal	float	float	cost		Cost of the restaurant
	internal	float	float	price		Price range of the restaurant
				cheap		
				expensive		
				average		
	internal	float	float	rating		Average rating of the restaurant
	internal	int	int	last_update		Last update of the restaurant
	internal	int	int	last_checked		Last checked of the restaurant

Structured and unstructured

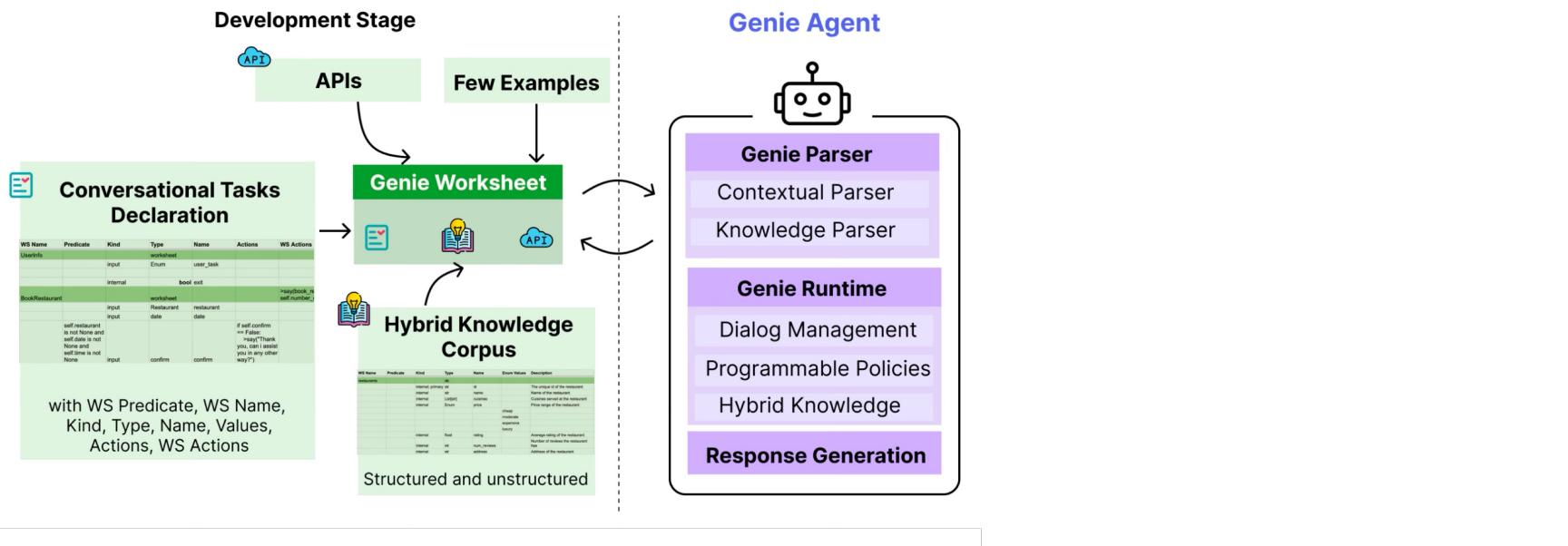
# Genie Worksheet

## A high-level declarative framework



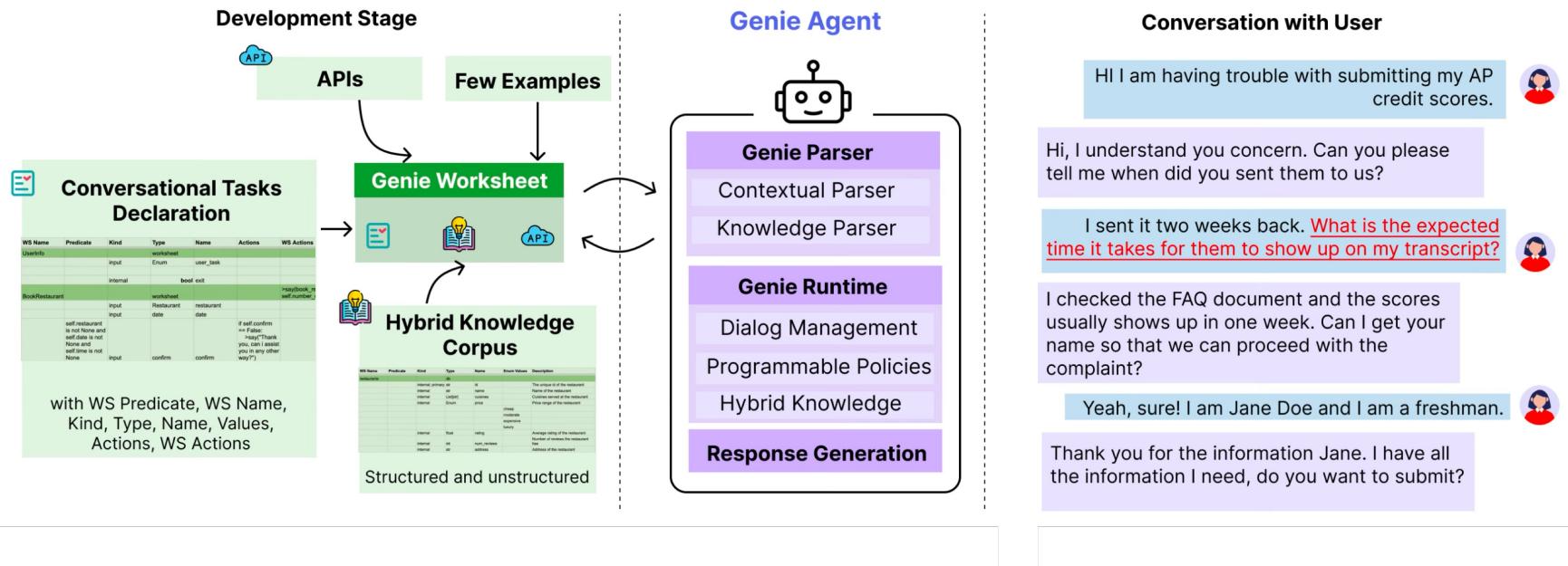
# Genie Worksheet

## A high-level declarative framework



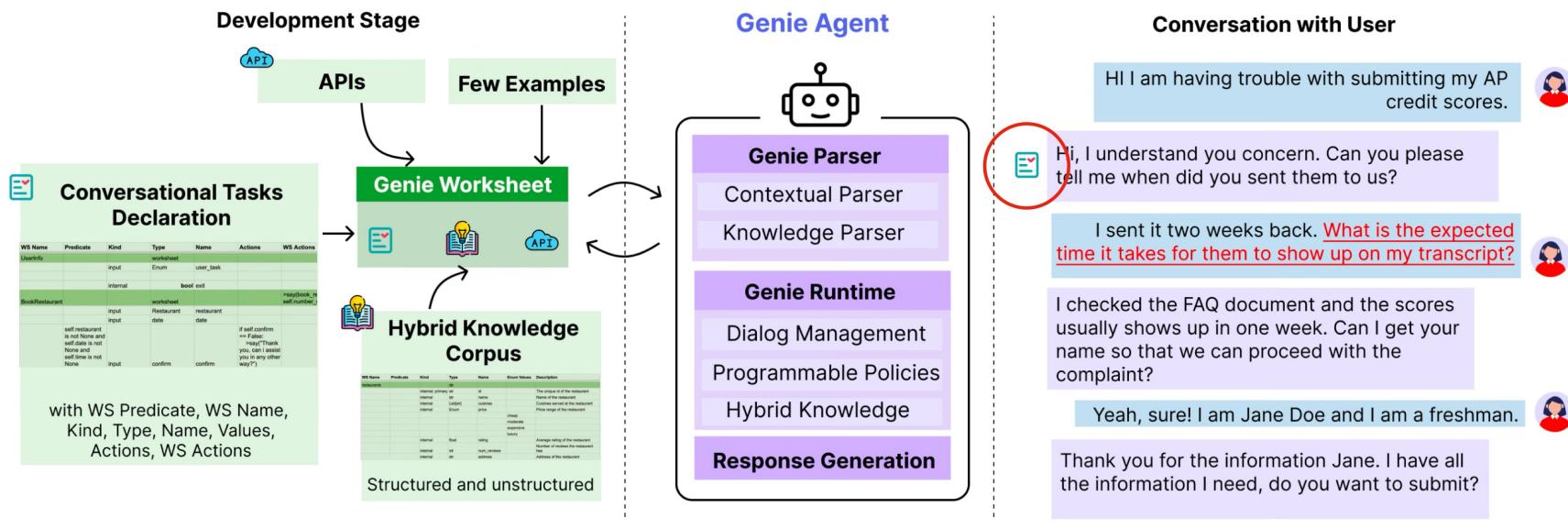
# Genie Worksheet

## A high-level declarative framework



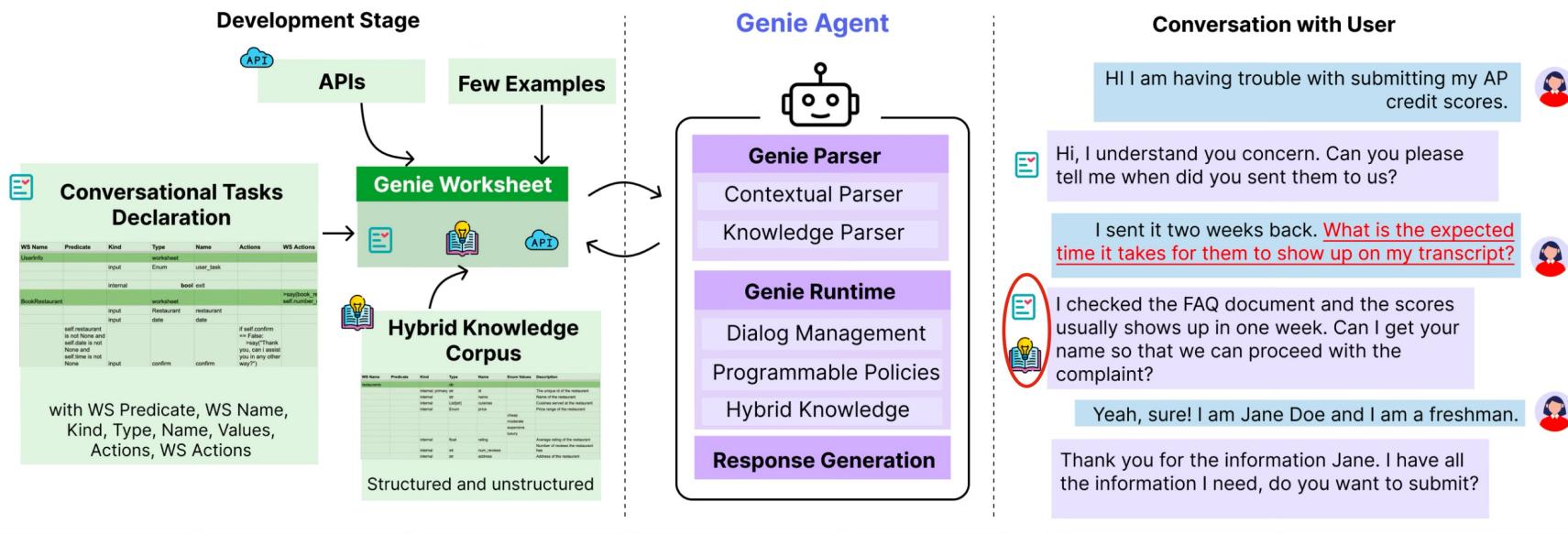
# Genie Worksheet

## A high-level declarative framework



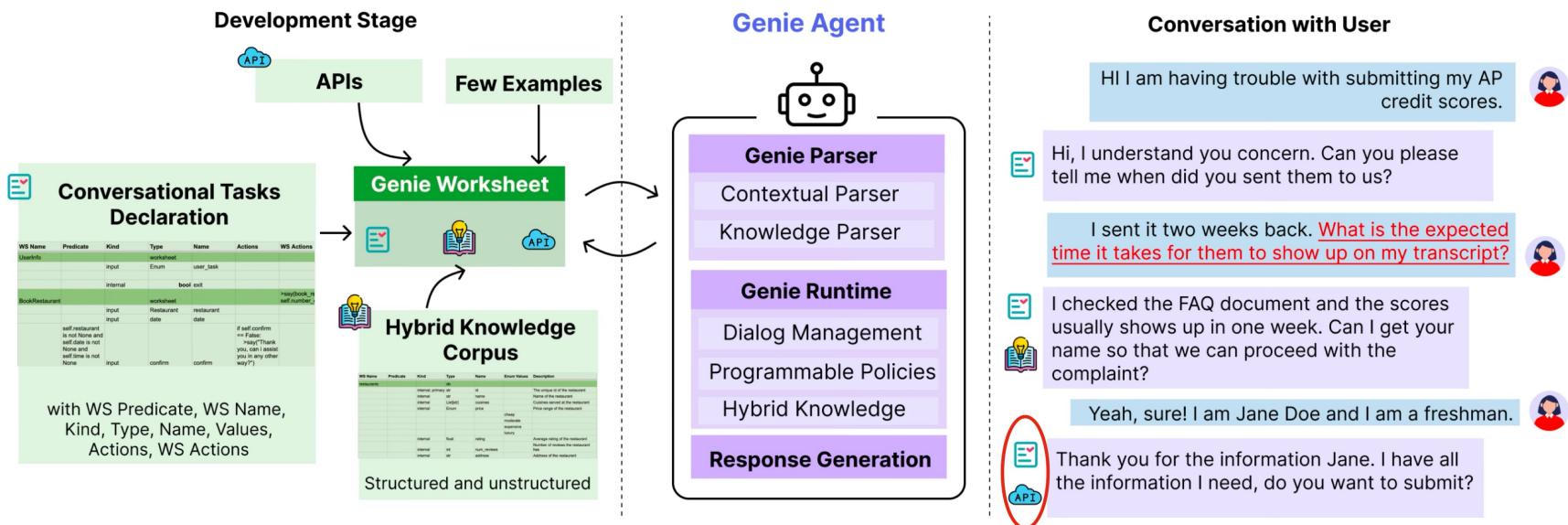
# Genie Worksheet

## A high-level declarative framework



# Genie Worksheet

## A high-level declarative framework



# Design of Genie Worksheets

- Fully Mixed-Initiatives agent
- High-level declarative specification:  
Worksheet + Knowledge Corpora
  - Worksheet capability: similar to webforms
  - Developer specifies the happy paths (agent initiatives)
  - Run-time system handles the unhappy paths
    - Using the knowledge resources given

Minimize coding effort

While creating an  
effective agent



# Agent Design Principles

- **Developer control:** Achieves the agent goals
- **Responsiveness:** User-initiatives are always given priority
- **Full, automatic support of knowledge queries:**  
Using SUQL framework
- **Expressiveness:**  
Supports arbitrary composition of knowledge queries and task statements
- **Effective for long conversations:**  
Via formal dialogue state tracking using Worksheets

# Mixed Initiatives

User  
Initiative

USER:

I want to take an AI course for 3 units and letter grade.

I can help you with enrolling in an AI course. Here are some top-rated AI courses: CS 224V, CS 224N, CS205. Which one would you like to take?

The Math Theory for ML looks interesting, I will enroll in it.

I am enrolling you in CS 205 for letter grade and 3 units, would you like to confirm your enrollment?

Yes!

Agent  
Initiative

Agent  
Initiative

# User initiatives are given priority

USER:

ASSISTANT

I want to take an AI course for 3 units and letter grade.

I can help you with enrolling in an AI course. Here are some top-rated AI courses: CS 224V, CS 224N, CS205. Which one would you like to take?

What is the schedule for CS 224V?

CS 224V is scheduled for Monday and Wednesday at 3:00pm

Would you like to take it?

Yes!

# Developer Control

- Letting developers exercise control without onerous efforts.
- Control over critical aspects of the agent's operation including:
  - the flow of conversations,
  - the timing of actions, and
  - the information elicited from users.

# Developer Control

**Student:** I want to take CS 224V in the fall quarter

**Agent:** Unfortunately, there are no seats available for CS 224V. However, I can find you courses similar to CS 224V which are offered in the fall quarter.

*Developer defined response*

**Student:** I want to take CS 224V in the fall quarter

**Agent:** Unfortunately, there are no seats available for CS 224V. However, you can ask for special permission if you are graduating soon!

*Developer defined response*

# Design of Genie Worksheets

# Task Worksheets

- Task Definition: Name, Predicate, Kind and Action
- Field Definition: Predicate, Name, Kind, Type, Description, Don't Ask, Required, Confirmation
- Semantics matter for Name and Description.
- Define action as python code to allow developer flexibility.
- Built-in actions: say, propose

# Design of Genie Worksheets

Ticket Submission Worksheet												
WS Predicate	WS Name	Predicate	Kind	Type	Name	Enum Values	Description	Don't Ask	Required	Confirmation	Actions	WS Actions
1	Main	Task	input	Enum	student_task	TroubleShoot Leave of Abs. Test Credits	The type of student requ...	TRUE			>say(submit_ticket(self.student_task, ...))	
	5	Fields can have worksheets as a type	input	Trouble Shoot	trouble_shoot		The enrollment issues that the student is facing	TRUE				
		self.student_task == "TroubleShoot"	input	Trouble Shoot	trouble_shoot	...						
		self.student_task is not None and ( self. .... )	input	str	extra_details		Ask for any other detail that the student wants to add	TRUE				
			input	confirm	confirm		Confirm that the student wants to submit the ticket.	TRUE			if self.confirm == False: >say("Thank you, how else can I help you?") >exitws()	
		self.student_task == "Trouble..."	TroubleShoot	WS	Two kinds of worksheets	3						
			services_general_info	KB								

# Knowledge Base Worksheets

- Real life queries often involve both structured and unstructured accesses.
- Use SUQL to handle hybrid knowledge queries
- Create a worksheet with Kind KB.

# Lecture Goals

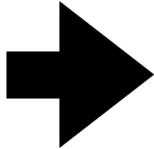
- Key concepts in task-oriented agents
- Prior Agent Policy Research
- Design and rationale for GenieWorksheets
- **Technical details of GenieWorksheets**
- Evaluating Genie Agents

# Specification to Python Dataclasses

 **Conversational Tasks Declaration**

WS Name	Predicate	Kind	Type	Name	Actions	WS Actions
Userinfo			worksheet			
	input		Enum	user_task		
	internal			bool exit		
BookRestaurant			worksheet			
	input		Restaurant	restaurant		
	input		Date	date		
	self.restaurant is not None and self.date is not None and self.time is not None				if self.confirm == False: confirm = say("Thank you, can i assist you in any other way?")	
	input		confirm	confirm		

with WS Predicate, WS Name, Kind, Type, Name, Values, Actions, WS Actions



```
1 def UserInfo(  
2     user_task: Enum["book_restaurant", "ask_information"]  
3 ):  
4     ...  
5  
6  
7 def BookRestaurant(  
8     restaurant: Restaurant,  
9     date: Date, num_people: int, confirm: Confirm  
10 ):  
11     ...  
12  
13  
14 class Restaurant:  
15     _id: int  
16     name: str  
17     location: Location
```

Translates specifications from Genie Worksheets into Python programs and executes them based on user utterances.

# Dialogue State Tracking with Worksheets

- LLMs struggle with maintaining relevant contextual information across longer conversation
- The Genie agent tracks the conversation **using a formal dialogue state**
- As an agent runs, the Genie Parser generates a set of changes to apply to the dialogue state

Formal Dialogue State ( $d_{t-1}$ )

```
answer_1 = Answer("show courses with light workload")
answer_1.result = [{course_id: "CS161"}, {course_id: "HCI347"}, ...]
course_1 = Course(course: answer_1.result)
course_to_take_1 = CourseToTake(course_0_details=course_0)
main_1 = Main(courses_to_take=course_to_take_1)
```

Formal Dialogue State ( $d_t$ )

```
answer_1 = Answer("""What is the max number of
units for HCI 347""",
"""SELECT max_units, course_id FROM courses
WHERE course_id ='HCI347'""")
answer_1.result = [{course_id:hci347,max_units:5}]
courses_to_take_1 = CoursesToTake(course=course_1)
course_1 = Course(course=CourseRecord("HCI347"))
```

# Records

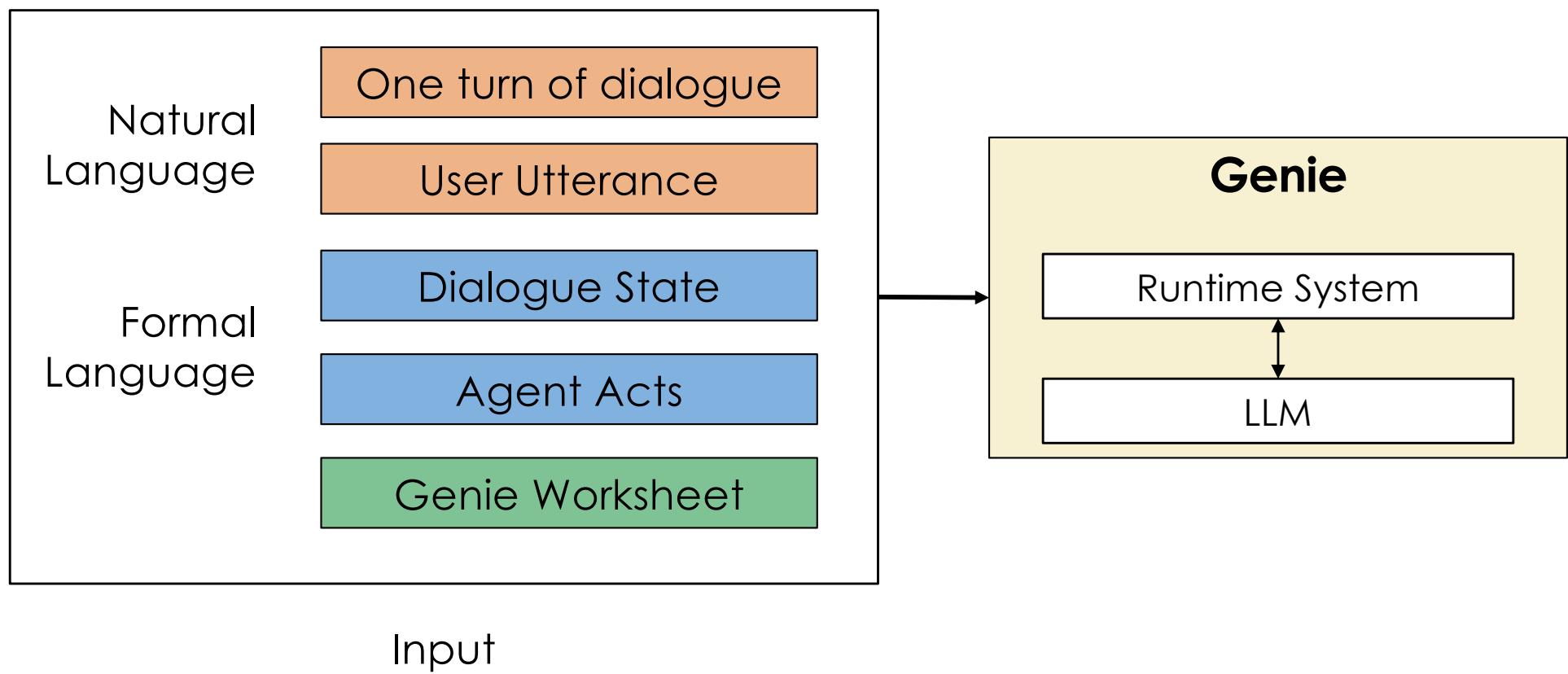
- Each record contains: Name, Kind
- Each task record contains: field names and values
- Each knowledge record contains: Natural language query, formal query, results

# Agent Acts

Five types of agent responses

- **Report:** Report the result by executing task or knowledge query in a record
- **Confirm:** Confirm value mapped to a field name in a record
- **Say:** Explicitly say the given utterance
- **Propose:** Propose a new task or query to the user
- **Ask:** Ask for the value of a field in a record

# Architecture Overview



# Natural Language Input

## Conversation History

**Student:** Can you suggest courses where students are happy with the load?

$u_{t-1}$

**Assistant:** I have found the following courses that where students are happy with the workload ....  
Which course would you like to take?

$a_{t-1}$

## User Utterance

**Student:** I will take the HCI course. Can I take it for more than 3 units?

$u_t$

- 1 turn of dialogue:  
Improves the semantic parsing performance
- Current user utterance

# Formal Language Input

Formal Dialogue State ( $d_{t-1}$ )

```
answer_1 = Answer("show courses with light workload")
answer_1.result = [{course_id: "CS161"}, {course_id: "HCI347"}, ...]
course_1 = Course(course: answer_1.result)
course_to_take_1 = CourseToTake(course_0_details=course_0)
main_1 = Main(courses_to_take=course_to_take_1)
```

Records: Given to the semantic parser as python code

Agent Acts ( $\bar{a}_t$ )

```
[Report(answer_1, answer_1.result), AskField(course_1,
course_units)]
```

Agent Acts

# One Conversational Turn

- Conversation History
- Formal Dialogue History
- Agent Acts
- Worksheet Specification

Conversation History

**Student:** Can you suggest courses where students are happy with the load?

$u_{t-1}$

**Assistant:** I have found the following courses that where students are happy with the workload .... Which course would you like to take?

$a_{t-1}$

User Utterance

**Student:** I will take the HCI course. Can I take it for more than 3 units?

$u_t$

Formal Dialogue State ( $d_{t-1}$ )

```
answer_1 = Answer("show courses with light workload")
answer_1.result = [{course_id: "CS161"}, {course_id: "HCI347"}, ...]
course_1 = Course(course: answer_1.result)
course_to_take_1 = CourseToTake(course_0_details=course_0)
main_1 = Main(courses_to_take=course_to_take_1)
```

Agent Acts ( $\bar{a}_{t-1}$ )

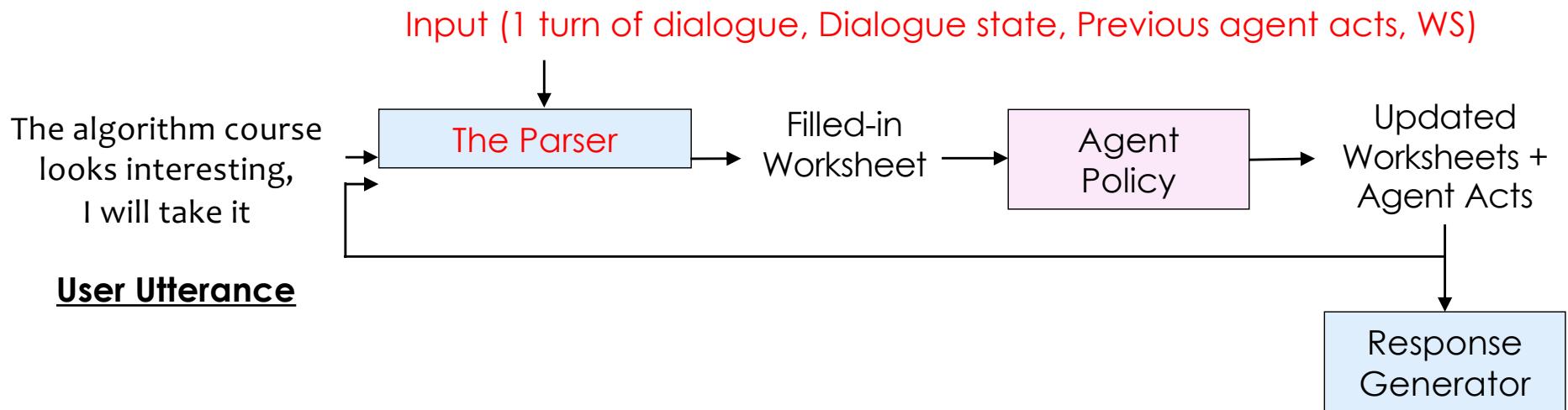
```
[Report(answer, answer.result), AskField(course, course_name)]
```

Worksheet Representation ( $w$ )

```
Answer(query: str)
Main(courses_to_take: CoursesToTake,
     student_details: StudentDetails,
     confirm: confirm)
CoursesToTake(course_0_details: Course,
             course_1_details: Course, ...)
StudentDetails(student_name: str, age: int,
              international_student: bool)
Course(course: CourseRecord)
```



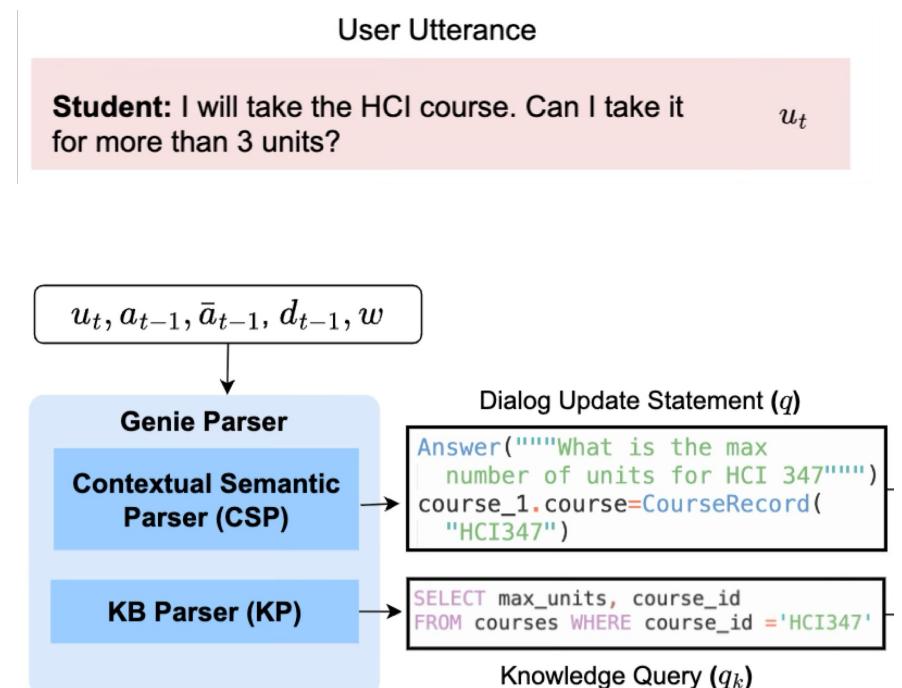
# Genie



# The Parser

## Three modules:

- Contextual Semantic Parser
  - Natural Language to Worksheet representation
- Knowledge Parser
  - Natural Language to SUQL query
- Update Module
  - Updates the dialogue state by executing code



# Contextual Semantic Parser

User can:

- Supply values to existing worksheet
- Modify previously filled field or remove its value
- Initiate a new task or query (generates a new record)

Latest Dialogue State ( $d_{t-1}$ )

courses\_to\_take\_1 = CoursesToTake(course=course)  
course\_1 = Course(course\_id="HCI347", units=3)

**User supplies value**

User: *I want to take it for letter grade!*  $u_t$

Dialogue statement update query  $q$ :

course\_1.grade\_type = "Letter"

**User modifies value**

User: *On second thoughts, i will take it for 4 units*  $u_t$

Dialogue statement update query  $q$ :

course\_1.units = 4

**User initiates task or query**

User: *I would also like to take CS 229 for credits.*  $u_t$

Dialogue statement update query  $q$ :

Course(course\_id="CS 229", grade\_type="Credits")

# Knowledge Parser

- Allows for more advanced approaches for parsing such as ReAct, Tree-of-thought.
- Abstracts knowledge integration from CSP

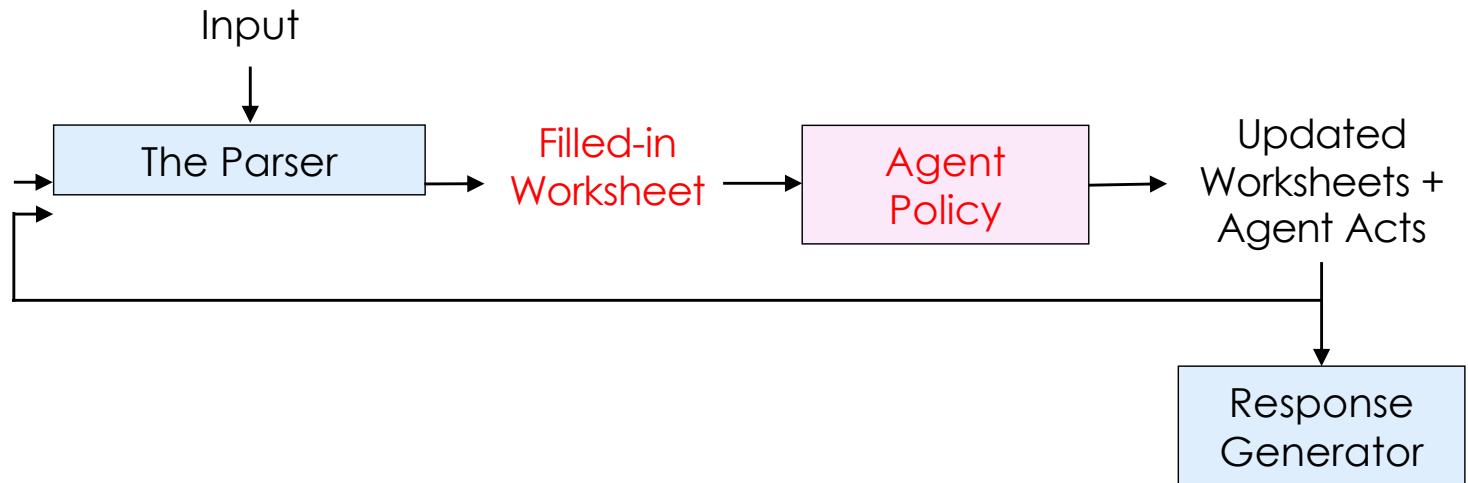


CSP only provides the knowledge query to the knowledge parser

# Genie

The algorithm curse  
looks interesting,  
I will take it

User Utterance



# Agent Policy

- LLM does not generate agent response.
- Genie uses a run-time system to compute the agent response

## **Genie:**

- runs knowledge queries and
- performs actions using a run-time system
- generates agent acts

# Agent Policy Rationale

- LLMs struggle with underrepresented developer-defined policies.
- LLMs cannot follow all the instructions.
- Agent acts forces the LLM to generate a **reliable responses** governed by the agent policy.

**Using an algorithmic module to compute the necessary actions improves the LLM's ability to follow instructions.**

# Run-Time System

## **The Parser:**

Assign variables to worksheet, generate database queries

**Agent Policy:** Output: Set of agent dialogue acts: A

Knowledge Query:

- If missing required parameter: Add “ask parameter” to A
- If query is complete, Evaluate query, Update variables

Worksheet:

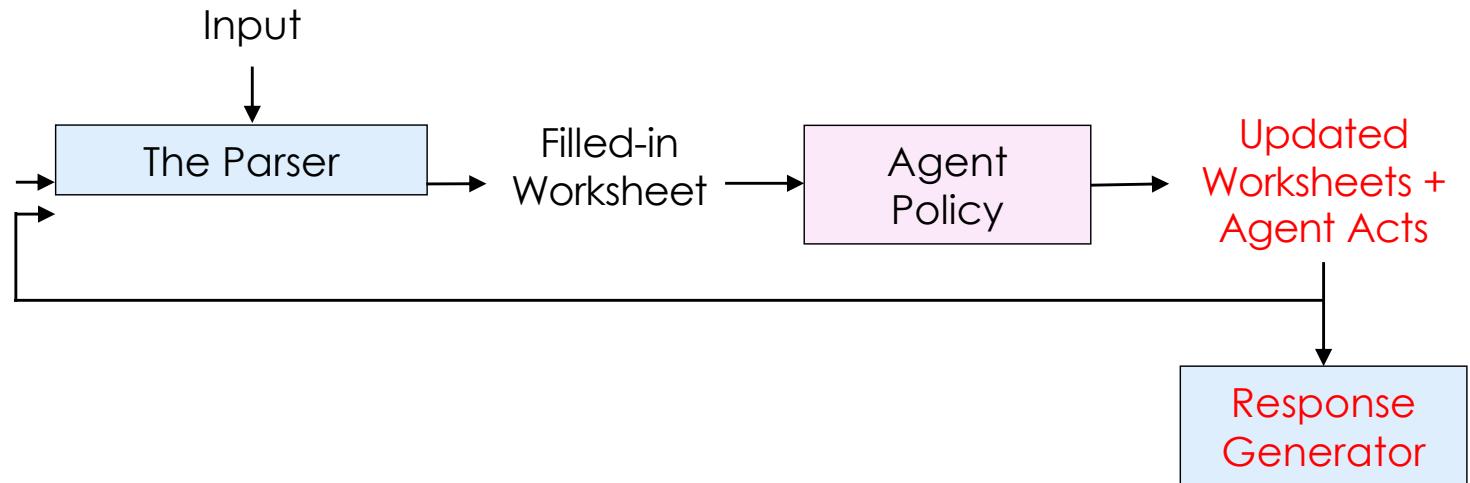
- If a field requires confirmation, Add “ask confirmation” to A or Execute
- If a WS is complete, Execute action
- If any required field is unfilled, Add “ask for field” to A

**Response Generation:** Generate response from A

# Genie

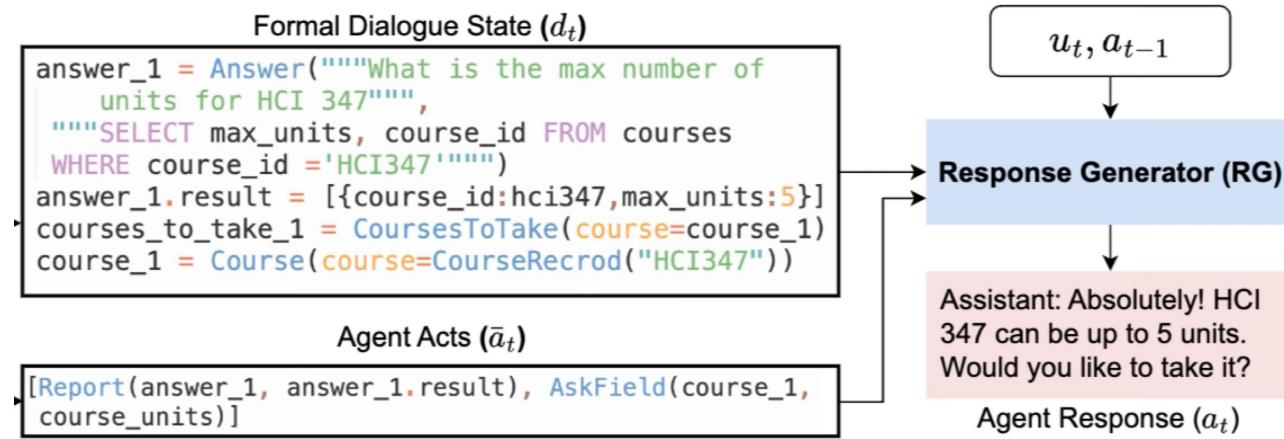
The algorithm curse  
looks interesting,  
I will take it

User Utterance



# Response Generator

- An LLM to convert the formal agent acts to natural language
- Dialogue state and the user utterance contextualizes the response



# Composition of Queries

- Compositions of queries and APIs are supported
- Passing the result of an instance of a worksheet in as a field of another worksheet.

User: I want to book an Italian restaurant in NYC for two on Valentine's day.

```
BookRestaurant(restaurant=Answer(SELECT * FROM restaurants  
WHERE 'italian' = ANY(cuisines) AND location = 'NYC;'),  
date="02/14/2024", and num_people=2)
```

# Complete Multiple tasks

USER:

ASSISTANT

I want to take an AI course for 3 units and letter grade.

I can help you with enrolling in an AI course. Here are some top-rated AI courses: CS 224V, CS 224N, CS205. Which one would you like to take?

# Data Dependent Fields

- Some fields are needed only for some users

Agent: Are you an international Student

User: yes, I am currently on a study visa.

Agent: What type of visa is it? F1 or J1?

# Support Multiple Knowledge Queries

USER:

ASSISTANT

Can you suggest some highly rated NLP courses? Also, when does the next semester start?

Answer("highly rated NLP courses", SELECT \* FROM courses WHERE answer("is this NLP course?", description) AND ratings > 4.5)

Answer("when does the semester start?", faq-free-text)

The Modeling Linguistic Complexities has a rating of 4.6.

According to the latest circular, the next semester starts on 18th August.

# Changing previous answers

USER:

ASSISTANT

I want to fly from New York to Los Angeles on 8th this month.

I have found 3 flights for you. ... Which one do you want to fly?

Oh wait, I have a meeting on 8th, I think I will leave on 9th.

# Genie Worksheets can support

- Allow data-dependent fields
- Complete multiple tasks
- Support multiple knowledge queries in a single utterance
- Keeping track of essential information
- Support task and knowledge requests in a single utterance
- Allow un-happy paths, eg. Changing answers

# Lecture Goals

- Key concepts in task-oriented agents
- Prior Agent Policy Research
- Design and rationale for GenieWorksheets
- Technical details of GenieWorksheets
- **Evaluating Genie Agents**

# Evaluation

- STAR Dataset
  - 13 domains and 22 tasks
  - Benchmarking Next Action Prediction
  - Provides policy graph outlining conversation flow
- STARV2 improves STAR dataset by updating:
  - belief state and action state annotations
  - More intuitive natural language descriptions for schema

# Evaluation Metrics

- User Action F1
  - Map user's natural language to user dialogue acts
- System Action F1
  - The next action the agent should take based on dialogue state and user dialogue acts
- Joint Goal Accuracy
  - Accuracy of the current dialogue state for any given turn

# Complex Program Logic in STARv2

- Bank, Trivia and Trip domains
- SOTA: AnyTOD
  - finetuned T5 (13B) on 6000 examples (all domains except the testbed)
  - Implement programs to handle logic

<b>Model</b>	<b>Bank</b>	<b>Trip</b>	<b>Trivia</b>
AT XXL	54.3	52.4	73.8
AT-SGD XXL	53.1	51.5	81.1
AT-PROG XXL	61	60.8	73.7
AT-PROG+SGD XXL	65	62.9	86.3

System Action F1

Is this good enough?

# Genie Agent on STARv2

- Define program logic in 9 lines of code
- Uses three examples for semantic parsing
- Most errors caused by inconsistent data annotations

Agent	Bank	Trip	Trivia
<i>Finetuned T5 (11B)</i>			
AT XXL	54.3	52.4	73.8
AT-SGD XXL	53.1	51.5	81.1
AT-PROG XXL	61.0	60.8	73.7
AT-PROG +SGD XXL	65.0	62.9	86.3
<i>Zeroshot</i>			
Llama 3.1 70B (FC)	48.9	41.7	81.7
GPT-4o-mini (FC)	50.8	43.8	69.8
GPT-4 Turbo (FC)	55.1	42.7	82.5
Genie + Llama 3.1 70B ( <b>Ours</b> )	82.1	75.9	82.2
Genie + GPT-4o-mini ( <b>Ours</b> )	<b>82.5</b>	<b>80.5</b>	<b>90.3</b>
Genie + GPT-4 Turbo ( <b>Ours</b> )	<b>82.5</b>	<b>83.4</b>	<b>92.7</b>

System Action F1

# Limitations of existing datasets

- Simple domains that require slot filling
- *Recall: function calling is not enough*
- Need to evaluate turn-wise performance
- Few “unhappy” paths in the dataset

# Evaluation with real users

Three diverse applications with varying complexities.

- **Restaurant Reservation:** Uses real-life database from Yelp
- **Ticket Submission:** Subset of service now api with several APIs
- **Course Enrollment:** Uses real-life database of courses at Stanford University and contains several fields.

Applications	Task WSs	KB WSs	Fields	Predicates	Actions
StarV2 (Bank)	3	0	10	4	4
StarV2 (Trip)	2	0	6	0	2
StarV2 (Trivia)	2	0	6	0	3
Restaurant Reservation	2	2	19	2	3
Course Assistant	4	4	52	3	1
Ticket Submission	7	1	29	18	2

# Baseline and Study Design

- Compare to GPT-4 turbo with Function Calling.
  - Uses the same KB-Parser as Genie Worksheet
- Recruited 22 and 20 users from Prolific for Restaurant Reservation and Ticket Submission
- Also, 20 university students to evaluate the course enrollment assistant
- Randomly assigned Genie Agent or GPT-4 (FC)

# Evaluation Metrics

- **Semantic Parsing Accuracy:**
  - Gold: for a given user's natural language query the set of API calls, Database calls, and fields to fill
- **Execution Accuracy:**
  - Check if the agent executes the correct API or database with the correct parameters
- **Agent Dialogue Accuracy:**
  - Measure of whether the agent follows the policies provided by the developer
- **Goal Completion Rate:**
  - User's ability to successfully complete the task using agent's assistance

# Results with Human Evaluation

- **Compressed context as formal dialogue state** enables LLM to invoke the correct API and database queries
- **Restaurant is common to academic datasets** that leads to better GA of GPT-4 (FC) compared to other domains
- GPT-4 FC often hallucinates non-existent courses despite being grounded in knowledge corpus.

	All Domains		Restaurant		Ticket Submission		Course Enrollment	
	GPT4 (FC)	Genie	GPT4 (FC)	Genie	GPT4 (FC)	Genie	GPT4 (FC)	Genie
SP Acc	-	91.4	-	93.8	-	85.8	-	94.1
Ex Acc	65.4	<u>86.5*</u>	50.0	<u>88*</u>	58.6	<u>80.0*</u>	79.2	<u>89.7*</u>
DA Acc	69.1	<u>89.2*</u>	57.7	<u>92.5*</u>	67.6	<u>82.5*</u>	77.8	<u>92.9*</u>
Goal CR	21.8	<u>82.8*</u>	54.5	<u>91.6</u>	0.0	<u>80.0*</u>	10.0	<u>80.0*</u>

LAM

STANFORD

# Conclusion

- LLMs need succinct context and fewer instructions to perform better in less known domains
- GenieWorksheets can help create **reliable and responsive** conversational agent by using:
  - formal dialogue state
  - runtime-system to execute agent policy
- Existing academic datasets are not enough for evaluation, need to run studies with real users.

# Projects

- AI-Powered Heart Failure Medication Assistant: Revolutionizing Patient Care Through Intelligent Conversation
- GenieWorksheet Wizard: Discovery of Missing Capabilities in Task-Oriented Agents from Simulation (Case study: Fidelity investments)
- Automatic Generation of a Dependable Conversational Agent from Real Conversations

# Homework

- Part 1: Interact with the fidelity investment agent built using Genie Worksheets
- Part 2: You will be creating an agent from the STARv2 dataset.