

PyDial

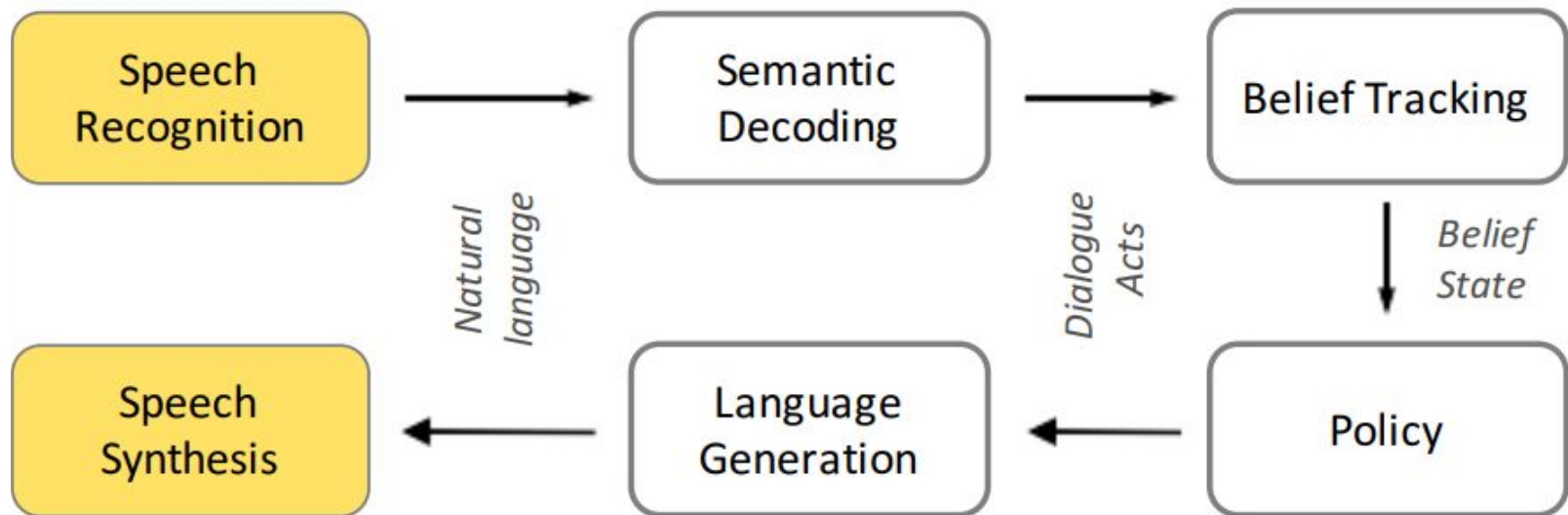
Multi-domain Statistical Dialogue System Toolkit

Developed by the Dialogue Systems Group at the University of Cambridge

Thang Vu, Chia-Yu Li, Glorianna Jagfeld, Daniel Ortega

Spoken Dialog System Architecture

2



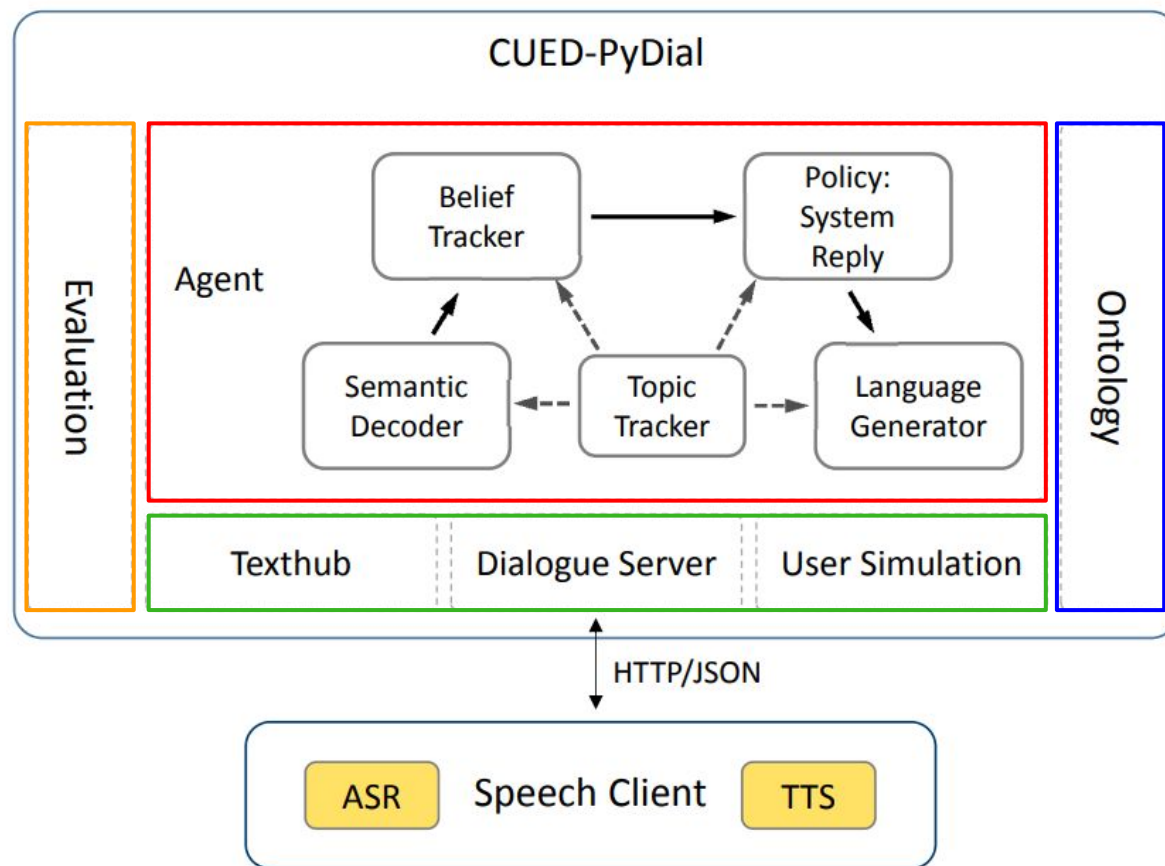
PyDial

3

- Is an open-source end-to-end statistical spoken dialogue system (SDS) toolkit
- Provides implementations of statistical approaches for all DS modules
- Provides multi-domain conversational functionality
- Key Principles:
 - Offers easy configuration
 - Easy extensibility
 - Domain-independent implementations

PyDial Architecture

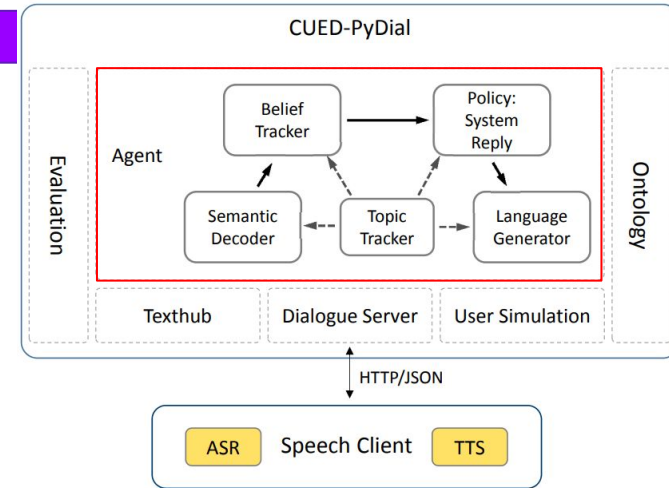
4



PyDial Components: Agent

5

- The **Agent** is responsible for the dialog interaction
- Agent's Modules:
 - **Semantic Decoder:** Text → Semantic representation
 - **Belief Tracker:** Maintains the internal dialog state representation (belief state)
 - **Policy:** Belief state → System dialogue act
 - **Language Generator:** System dialog act → Textual representation
 - **Topic Tracker:** Allows the multi-domain functionality
- The Agent maintains multiple sessions by ensuring that each input is routed to the correct dialog



PyDial Components: Interfaces

6

- **User Simulation**

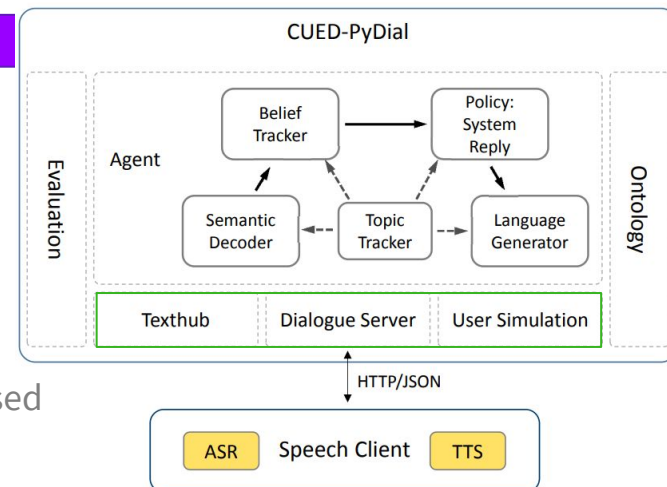
- Provides simulation of dialogs on the semantic level
- The Semantic Decoder and the Language Generator are not used

- **Text Hub**

- Connects the Agent to a terminal (This is the mode, we will use in our course)

- **Dialog Server**

- Enables the speech-based dialog mode via an external **Speech Client**
- The client maps:
 - Input speech → Text (Automatic Speech Recognition - ASR)
 - Output text → Speech (Text to Speech - TTS)
- HTTP connection exchanging JSON messages
- The Speech Client is **not** part of PyDial



PyDial Components: Ontology and Evaluation

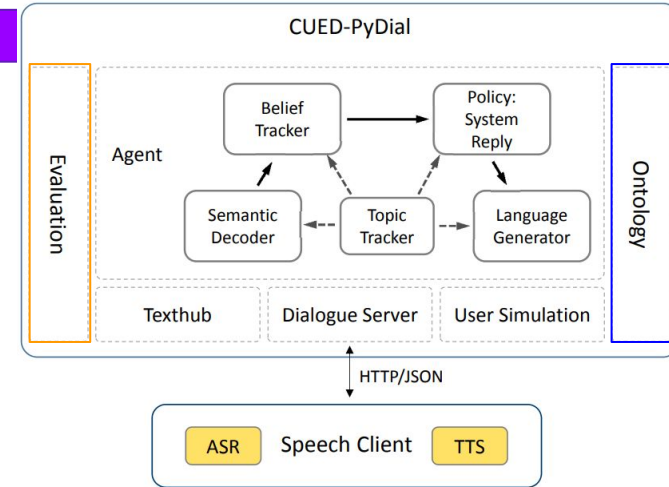
7

- **Ontology**

- Encapsulates the dialog domain specification
- Encapsulates the access the back-end database
- Is used by most of the DS modules and the User Simulator
- Provides useful information:
 - User actions, slots, slot values, system actions

- **Evaluation**

- Computes evaluation measures for dialogs



Multi-domain dialog System Architecture

8

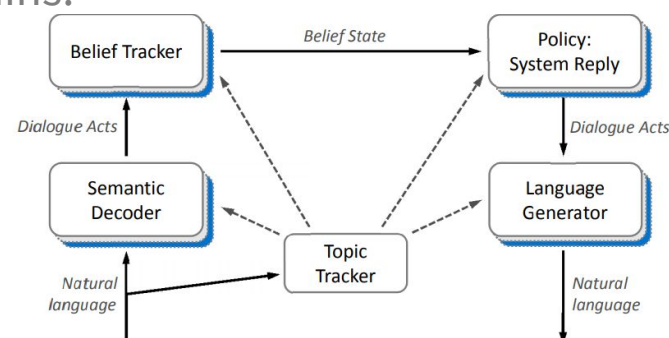
Pydial enables conversations ranging over multiple domains.

- **Assumptions**

- Each user input belongs to exactly one domain
- Only the user can switch domains

- **Topic Tracker**

- Identifies the domain the user input or the current sub-dialog belongs to
- Domain-specific instances of each dialog module are loaded based on the identified domain.



To handle the various domain instances, every module has a **Manager** which stores all of the domain-specific instances.

These instances are only created once for each domain and each agent.

PyDial: Key Principles

9

- **Domain Independence**

- The main functionality is domain independent by simply using a different domain specification, simulated dialogues using belief tracker and policy
- The Ontology handles all domain-related functionality and is accessible system-wide
- **Exception:** The Semantic Decoder and the Language Generator are domain-dependent

- **Easy Configurability**

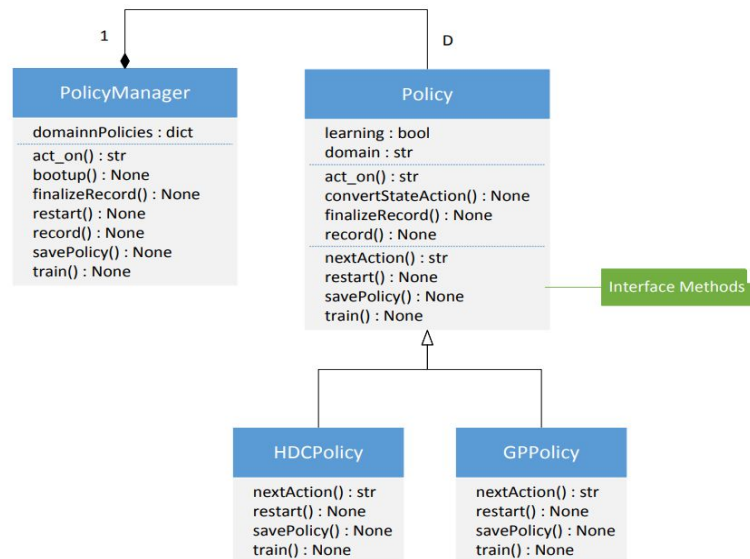
- All relevant functionality can be controlled via configuration files
- Domains of the conversation, the variants of each domain module and the parameters are specified

PyDial: Key Principles

10

- **Extensibility**

- The Manager allows easy extensibility
- The corresponding interface class must be implemented
 - Topic Tracker, Semantic Decoder
 - Belief Tracker, Policy
 - Language Generator and Evaluation
- A new domain is defined by a new ontology and a DB



PyDial: Built-in implementation

11

- **Semantic Decoder**
 - Rule-based approach
 - Statistical model based on Support Vector Machines
- **Belief Tracker**
 - Rule-based approach
- **Policy**
 - Hand-crafted policy
 - Gaussian process reinforcement learning policy
- **Language Generator**
 - Template-based language generator
 - LSTM-based language generator
- **Topic Tracker**
 - Keyword-based topic tracker
- **Evaluation**
 - Objective task success evaluator
 - Subjective task success evaluator
- **User Simulation**
 - Agenda-based user simulator

PyDial: Domains

12

- PyDial focus:
 - The user has to find a matching entity based on a number of constraints
 - Once the entity is found, the user can request additional information

Domain	Constraints [*]	Requests [†]	Entities
CamRestaurants	3	9	110
CamHotels	5	11	33
CamAttractions	3	9	79
CamShops	2	6	21
CamTransport	2	6	5
SFRestaurants	6	11	271
SFHotels	6	10	182
Laptops6	6	10	123
Laptops11	11	21	123
TV	6	14	94

^{*} # data base search constraints, [†] # information the user can request

PyDial: Example

13

```
bash$ python pydial.py chat config/Tut-hdc-CamInfo.cfg
*** Chatting with policies ['CamRestaurants', 'CamHotels']: ***
Prompt > Hello, welcome to the Cambridge Multi- Domain dialogue system. How may I help you?
User   > Hi! I am looking for a restrnt in the south of the city
Prompt > I' m having trouble understanding what you want. Do you want a Cambridge Restaurant, or a Cambridge Hotel?
User   > Restaurant
Prompt > What kind of food would you like?
User   > German
Prompt > What part of town do you have in mind?
User   > South
Prompt > I am sorry but there is no place, serving german food and in the area of south
User   > What about Mexican?
Prompt > chiquito restaurant bar is a nice place, it serves mexican food and it is in the south part of town .
User   > Oh, thanks! Could you tell me the address and the telephone number?
Prompt > chiquito restaurant bar is a nice place, Their address is 2g cambridge leisure park cherry hinton road cherry hinton and Their phone number is 01223 400170 .
User   > What is the price range?
Prompt > chiquito restaurant bar is a nice place it is in the expensive price range .
User   > Bye!
Prompt > chiquito restaurant bar is a nice place Their postcode is c.b 1, 7 d.y .
User   > Bye!
Prompt > chiquito restaurant bar is a nice place it is described as chiquito is a chain of inexpensive tex mex restaurants predominantly aimed at tourists. they serve a variety of mexican and north of the border food as well as a range of cocktails .
User   > Bye
Prompt > Thank you, goodbye.
```

The diagram illustrates the PyDial system's ability to extract specific information from a dialogue. Two purple arrows point from labels to specific parts of the text:

- A purple arrow labeled "Matching Entity" points from the text "chiquito restaurant" in the prompt to the word "chiquito" in the user's response.
- A purple arrow labeled "Requested Information" points from the text "address and the telephone number" in the user's response to the corresponding information provided in the prompt: "Their address is 2g cambridge leisure park cherry hinton road cherry hinton and Their phone number is 01223 400170 ."

References

14

- Deep Learning for Dialogue Tutorial @ACL 2017
<https://sites.google.com/site/deeplearningdialogue/home>
- Speech and Language Processing (Jurafsky, Martin) (3rd ed. draft)
<https://web.stanford.edu/~jurafsky/slp3/29.pdf>
- Stefan Ultes, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Pawel Budzianowski, Nikola Mrksić, Tsung-Hsien Wen, Milica Gasic, and Steve Young. PyDial: A Multidomain Statistical Dialogue System Toolkit. In Proceedings of ACL 2017, System Demonstrations, pages 73–78, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-4013>.