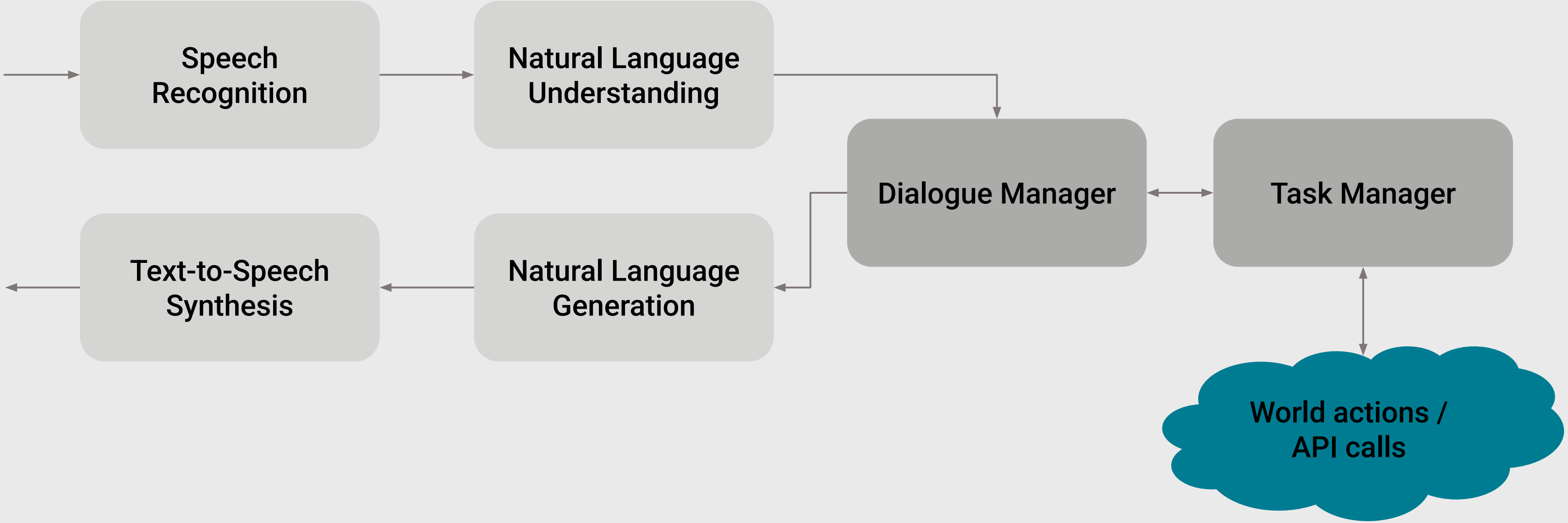
CS 224S / Linguist 285
Spoken Language Processing

Andrew Maas | Stanford University | Spring 2025

Lecture 8: Developing spoken dialog systems. Example dialog frameworks.

Today: Particular formalisms and frameworks for dialog

Tour of modern frameworks (course projects can use these!)



Speech Recognition

Natural Language Understanding

Dialogue Manager

Task Manager

Text-to-Speech Synthesis

Natural Language Generation

World actions / API calls

Outline

- Problem space: What type of conversational agent are you building?
- Foundational dialog management architectures
- Frame-based dialog systems
- Dialog framework examples and trade-offs

The space of possible dialog systems

What does the system need to do?

What types of interactions will it support?

Conversational Agent Problem Space

- **Time to response (Synchronous?)**
- **Task complexity**
  - What time is it?
  - Book me a flight and hotel for vacation in Greece
- **Interaction complexity / number of turns**
  - Single command/response
  - “I want new shoes” What kind? What color? What size?
- **Initiative**
  - User, System, Mixed
  - Interaction modality
- **Purely spoken, Purely text, Mixing speech/text/media**
- **Think about the interaction needs for a particular task/system. Implementation is very different depending on what point in the problem space you target**

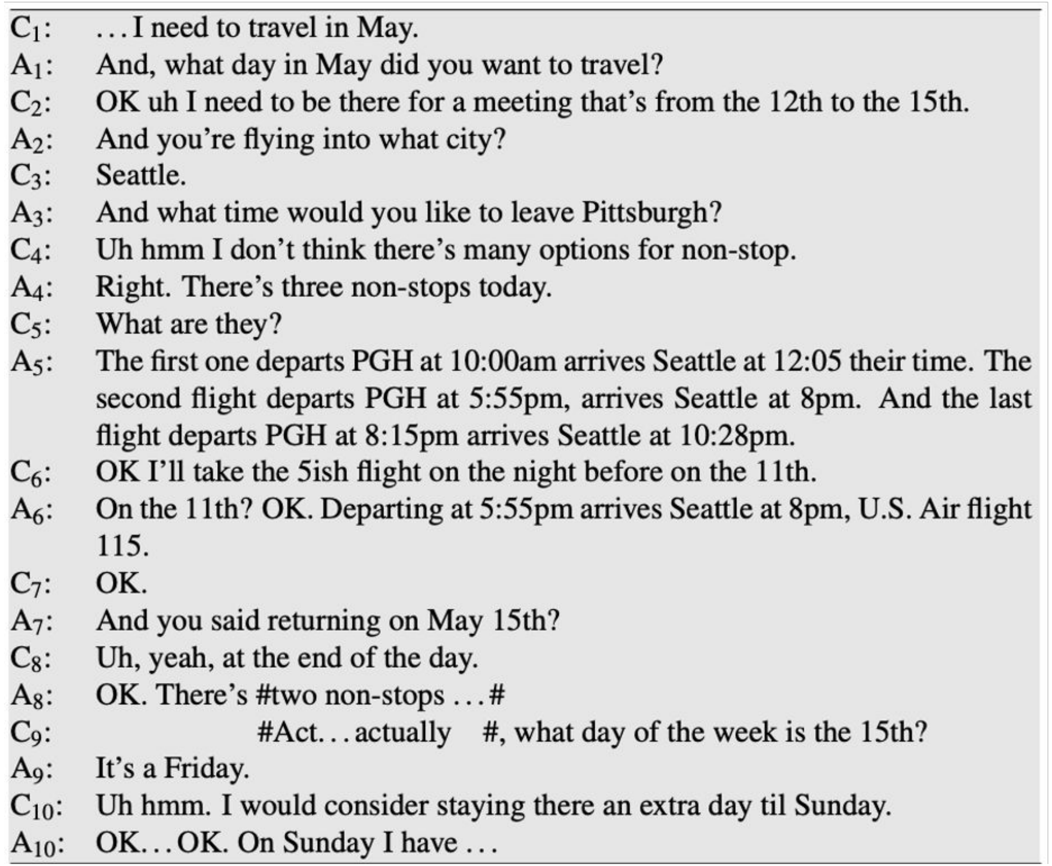
System Design Considerations

- Not all systems require support for complex interactions (sometimes voice commands work fine)
- Frameworks like Alexa force some choices about multiple modules to simplify overall development
- ASR/TTS components often be treated as black-box, but great systems are sensitive to ASR uncertainty, interruptions, and provide backchannel responses
- Okay to redefine/combine modules based on problem (e.g. a smart NLG module might simplify dialog manager)

- Big modern questions in design:
    - Focus on the user experience. What dialog interactions do you need to support? Synchronous? Etc.
    - Once UX defines requirements. What is the task/action space? (what actions the system can take in the world)
    - Design the conversational system to bridge between the UX and task outputs you want to support

Example task-oriented dialog

- Turn-taking
- Speech Acts
- Grounding
- Inference

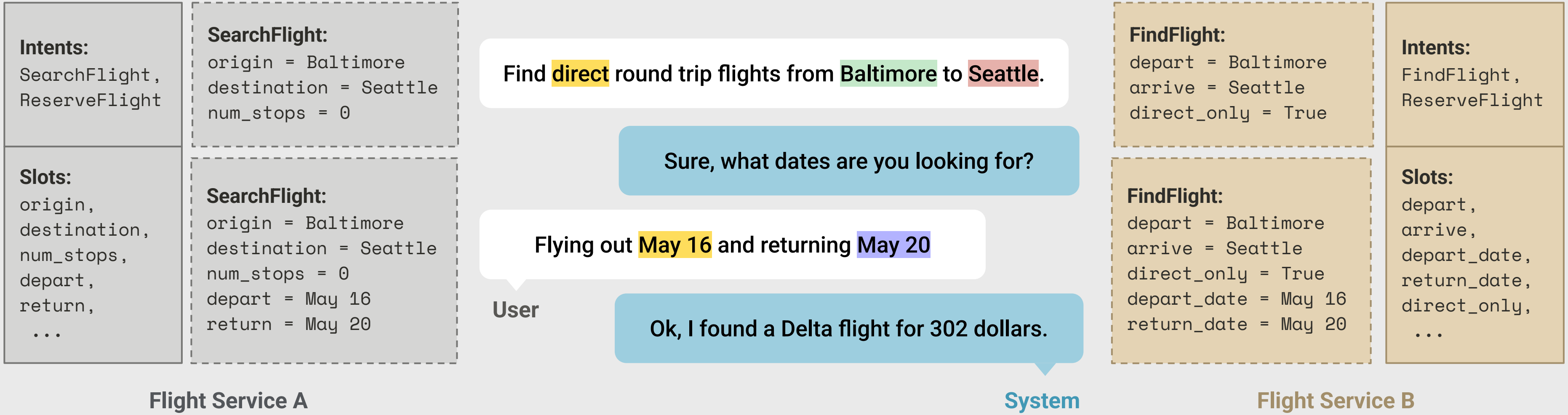


C_1: . . . I need to travel in May.
A_1: And, what day in May did you want to travel?
C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_2: And you're flying into what city?
C_3: Seattle.
A_3: And what time would you like to leave Pittsburgh?
C_4: Uh hmm I don't think there's many options for non-stop.
A_4: Right. There's three non-stops today.
C_5: What are they?
A_5: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
C_6: OK I'll take the 5ish flight on the night before on the 11th.
A_6: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
C_7: OK.
A_7: And you said returning on May 15th?
C_8: Uh, yeah, at the end of the day.
A_8: OK. There's #two non-stops . . . #
C_9: #Act. . . actually #, what day of the week is the 15th?
A_9: It's a Friday.
C_{10}: Uh hmm. I would consider staying there an extra day til Sunday.
A_{10}: OK. . . OK. On Sunday I have . . .

Figure 2: Part of a phone conversation between a human travel agent (A) and human client (C). The passages framed by # in A_8 and C_9 indicate overlaps in speech

Dialog State Tracking (DST) as a task

Predict state (slot values + dialog act) from transcript + context. *Task data schema is a design choice*



Intents:
SearchFlight,
ReserveFlight

Slots:
origin,
destination,
num_stops,
depart,
return,
...

SearchFlight:
origin = Baltimore
destination = Seattle
num_stops = 0

SearchFlight:
origin = Baltimore
destination = Seattle
num_stops = 0
depart = May 16
return = May 20

Flight Service A

Find direct round trip flights from Baltimore to Seattle.

Sure, what dates are you looking for?

Flying out May 16 and returning May 20

User

Ok, I found a Delta flight for 302 dollars.

System

FindFlight:
depart = Baltimore
arrive = Seattle
direct_only = True

FindFlight:
depart = Baltimore
arrive = Seattle
direct_only = True
depart_date = May 16
return_date = May 20

Intents:
FindFlight,
ReserveFlight

Slots:
depart,
arrive,
depart_date,
return_date,
direct_only,
...

Flight Service B

Figure: Dialogue state tracking labels after each user utterance in a dialogue in the context of two different flight services. Under the schema-guided approach, the annotations are conditioned on the schema (extreme left/right) of the underlying service). [DSTC8 overview]

Emerging state of the art implementation approach

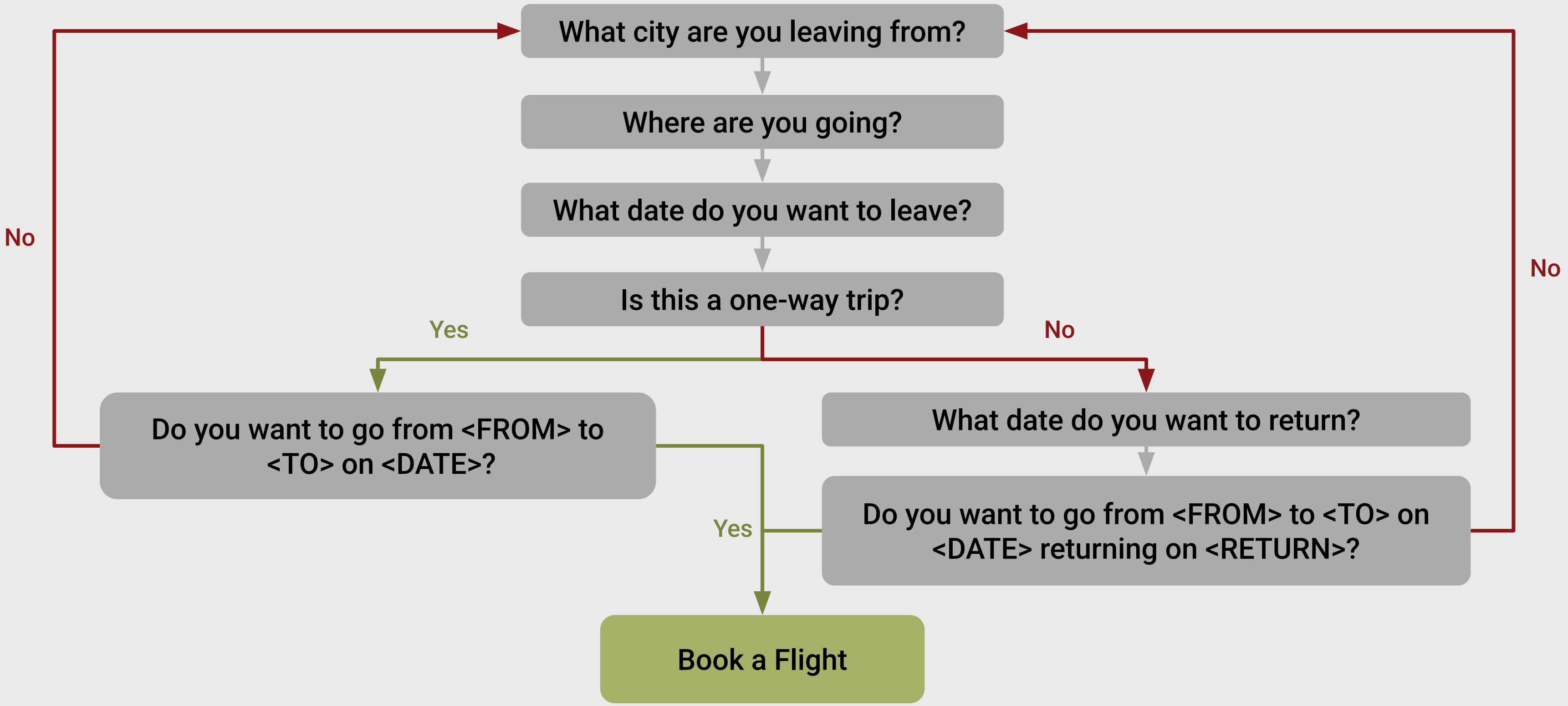
- **Use deep learning approaches for ASR/TTS. Keep these modules separate at first**
- **Use LLM for NLU and NLG**
  - Create custom markup representation for slots and their values in both NLU and NLG
  - Fine tune LLM for your tasks once initial system is stable
- **Dialog control + state tracking: Use task-specialized, smaller LLMs (e.g. dialog state, DB queries)**
  - LLM can guess about next action to take
  - Track semi-structured representation of dialog so far using LLM to update state
  - Interface with tasks / actions / external APIs using structured output from LLM
- **Training options (depends on availability):**
  - Use supervised learning to optimize per-task outputs
  - Interact with live/simulated users for reinforcement learning
  - Specialize NLU/NLG and ASR/TTS to domain-specific vocabulary for your task

Foundational dialog management architectures

Possible Architectures for Dialog Management

- **Finite State**
    - Resurgence in utility because LLM-based modules allow simple state machines to do a lot overall
- **Frame-based**
    - Alexa skills kit uses a version of this
- **Information State (Markov Decision Process)**
    - Facilitates reinforcement learning formulation for dialog control. Some similarities w/ state machines
- **LLM / deep learning systems**
    - Active research area, no single approach. More lectures on this later in quarter

Finite State Dialog Manager


What city are you leaving from?
Where are you going?
What date do you want to leave?
Is this a one-way trip?
Yes
No
Do you want to go from <FROM> to <TO> on <DATE>?
What date do you want to return?
Do you want to go from <FROM> to <TO> on <DATE> returning on <RETURN>?
No
No
Yes
Book a Flight

Finite-State Dialog Managers

- System completely controls the conversation with the user.
- It asks the user a series of questions
- Ignoring (or misinterpreting) anything the user says
  that is not a direct answer to the system's questions
- Quick solution for simple tasks, scales poorly to complex/large tasks
- Consider a trivial airline travel system. State enumeration becomes unwieldy:
  - Ask the user for a departure city
  - Ask for a destination city
  - Ask for a time
  - Ask whether the trip is round-trip or not

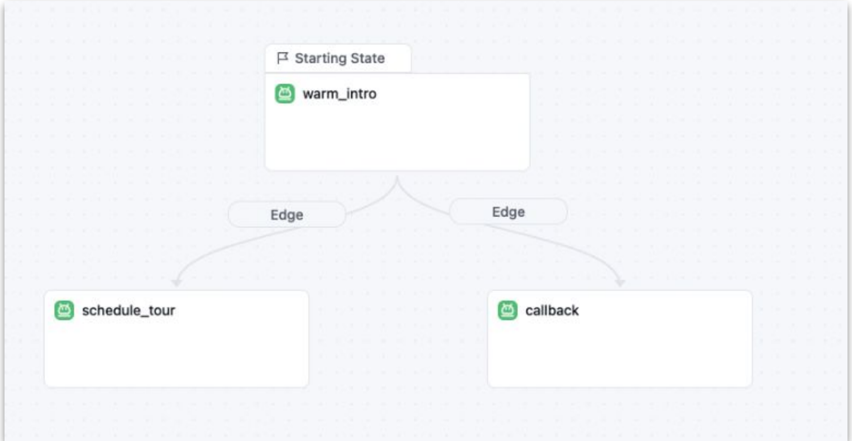
Exciting new thing: State machines + LLM prompt states

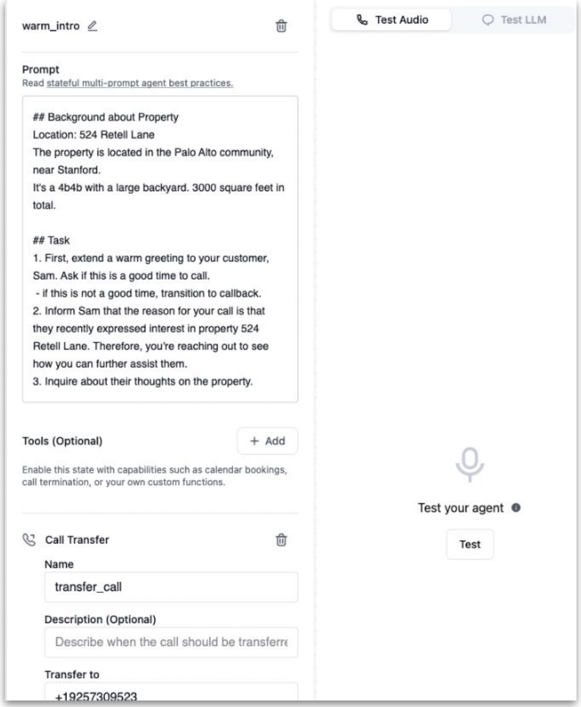
Few states with powerful LLM-based module at each state

High-level state machines are manageable to build, allow enforcing top-level concepts or flow

LLM-based systems means each state can be a context-specific LLM (+API / context population)

Example: Retell AI dialog flow builder





warm_intro

Prompt
Read stateful multi-prompt agent best practices.

Background about Property
Location: 524 Retell Lane
The property is located in the Palo Alto community, near Stanford.
It's a 4b4b with a large backyard. 3000 square feet in total.

Task
1. First, extend a warm greeting to your customer, Sam. Ask if this is a good time to call.
 - if this is not a good time, transition to callback.
2. Inform Sam that the reason for your call is that they recently expressed interest in property 524 Retell Lane. Therefore, you're reaching out to see how you can further assist them.
3. Inquire about their thoughts on the property.

Tools (Optional)

Enable this state with capabilities such as calendar bookings, call termination, or your own custom functions.

Call Transfer

Name: transfer_call

Description (Optional): Describe when the call should be transferre

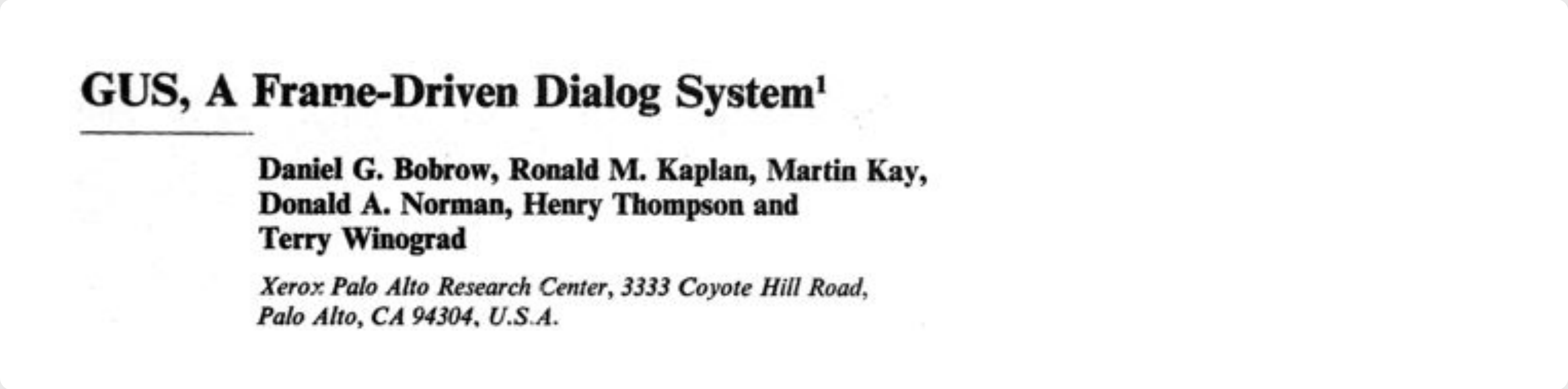
Transfer to: +19257309523

Test your agent

GUS and Frame-Based Systems

Frame-Based Dialogue Agents

- 1977



GUS, A Frame-Driven Dialog System[1]

Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay,
Donald A. Norman, Henry Thompson and
Terry Winograd

Xerox Palo Alto Research Center, 3333 Coyote Hill Road,
Palo Alto, CA 94304, U.S.A.

- Still the industrial state of the art (LLM-based systems not reliable or controllable enough yet)
- Alexa roughly follows GUS architecture

The Frame

- A set of **slots**, to be filled with information of a given **slot type**
- Each associated with a **question** to the user

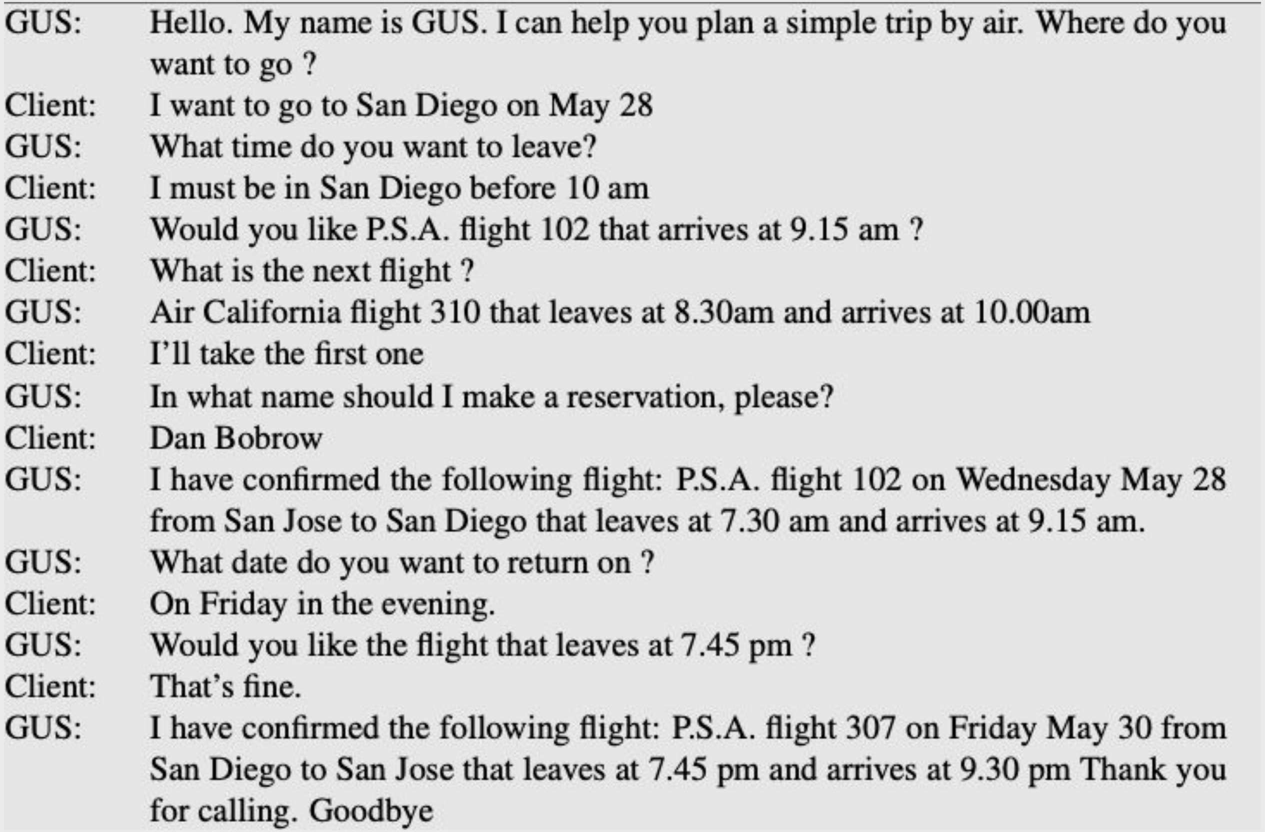
Slot	Type	Question
ORIGIN	city	What city are you leaving from?
DEST	city	Where are you going?
DEP DATE	date	What day would you like to leave?
DEP TIME	time	What time would you like to leave?
AIRLINE	line	What is your preferred airline?

Frame-based data abstractions

- One or more **frames. A frame specifies:**
    - A collection of **slots**
    - Each slot having a **value** of some **slot type**
    - Required slots, other types and constraints of their values, and any inter-slot dependencies (e.g., departure date must precede return date). Might also imply data value checking operations

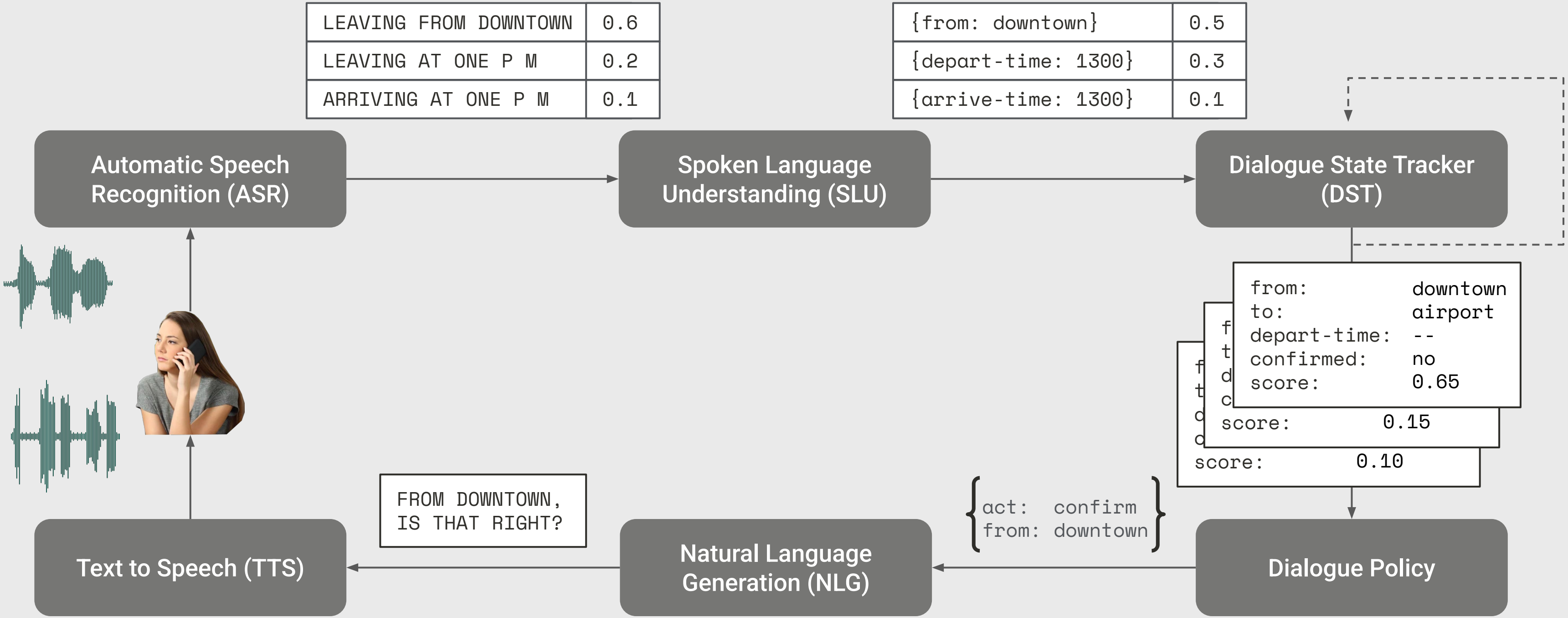
- **Domain ontology** is full collection of frames for a (sub)system
    - Complete inventory of frames and their associated slot definitions for a given subsystem, serving as the dialogue system's schema-level knowledge base
    - Decouples domain knowledge from control logic. Allows extending the system with new frames for additional subtasks without rewriting core dialogue-management code. *This is often a fundamental goal for dialog frameworks.*

The State of the Art in 1977



GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go ?
Client: I want to go to San Diego on May 28
GUS: What time do you want to leave?
Client: I must be in San Diego before 10 am
GUS: Would you like P.S.A. flight 102 that arrives at 9.15 am ?
Client: What is the next flight ?
GUS: Air California flight 310 that leaves at 8.30am and arrives at 10.00am
Client: I'll take the first one
GUS: In what name should I make a reservation, please?
Client: Dan Bobrow
GUS: I have confirmed the following flight: P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.
GUS: What date do you want to return on ?
Client: On Friday in the evening.
GUS: Would you like the flight that leaves at 7.45 pm ?
Client: That's fine.
GUS: I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm Thank you for calling. Goodbye

Dialog pipeline with explicit frame data structures



LEAVING FROM DOWNTOWN	0.6
LEAVING AT ONE P M	0.2
ARRIVING AT ONE P M	0.1

{from: downtown}	0.5
{depart-time: 1300}	0.3
{arrive-time: 1300}	0.1

Automatic Speech Recognition (ASR)
Spoken Language Understanding (SLU)
Dialogue State Tracker (DST)
Dialogue Policy
Natural Language Generation (NLG)
Text to Speech (TTS)

```
from:          downtown
to:            airport
depart-time:   --
confirmed:     no
score:         0.65
```

score: 0.15

score: 0.10

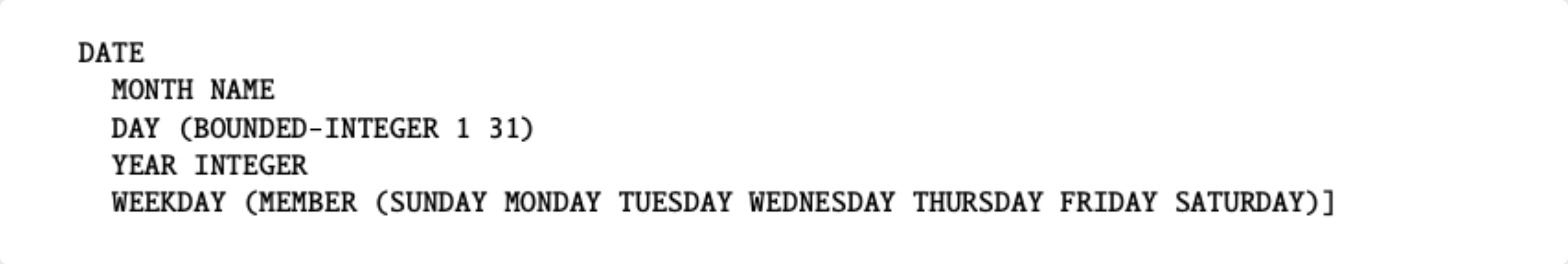
```
act:   confirm
from:  downtown
```

FROM DOWNTOWN,
IS THAT RIGHT?

Figure: Architecture of dialogue-state system for task-oriented dialogue (William et al, 2016)

Slot Types Can Be Complex, Hierarchical

- The type DATE:



```
DATE
  MONTH NAME
  DAY (BOUNDED-INTEGER 1 31)
  YEAR INTEGER
  WEEKDAY (MEMBER (SUNDAY MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY)]
```

- Generalization: Nested JSON or similar data structures with primitive types and data classes

Frames and mixed initiative dialog for slot filling

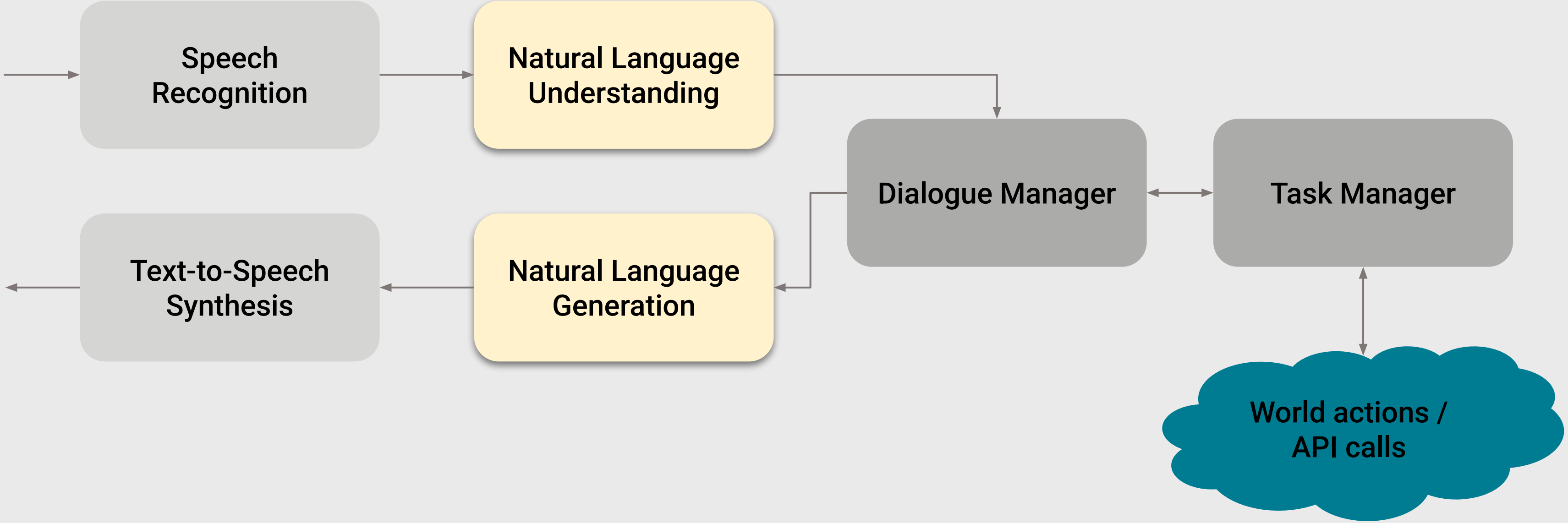
- System asks questions of user, filling any slots that user specifies
  - When frame is filled, do database query / API call
- If user answers 3 questions at once, system can fill 3 slots and not ask these questions again!
- Frame structure guides dialog. Frame data structure completion implies what to ask next
  - Conversational initiative can shift between system and user while focusing on frame completion

Slot	Type	Question
ORIGIN	city	What city are you leaving from?
DEST	city	Where are you going?
DEP DATE	date	What day would you like to leave?
DEP TIME	time	What time would you like to leave?
AIRLINE	line	What is your preferred airline?

NLU and NLG in frame-based systems

LLMs allow NLU + NLG modules to be more flexible

Broader scope, more robust, complex language/data mapping


Speech Recognition

Natural Language Understanding

Dialogue Manager

Task Manager

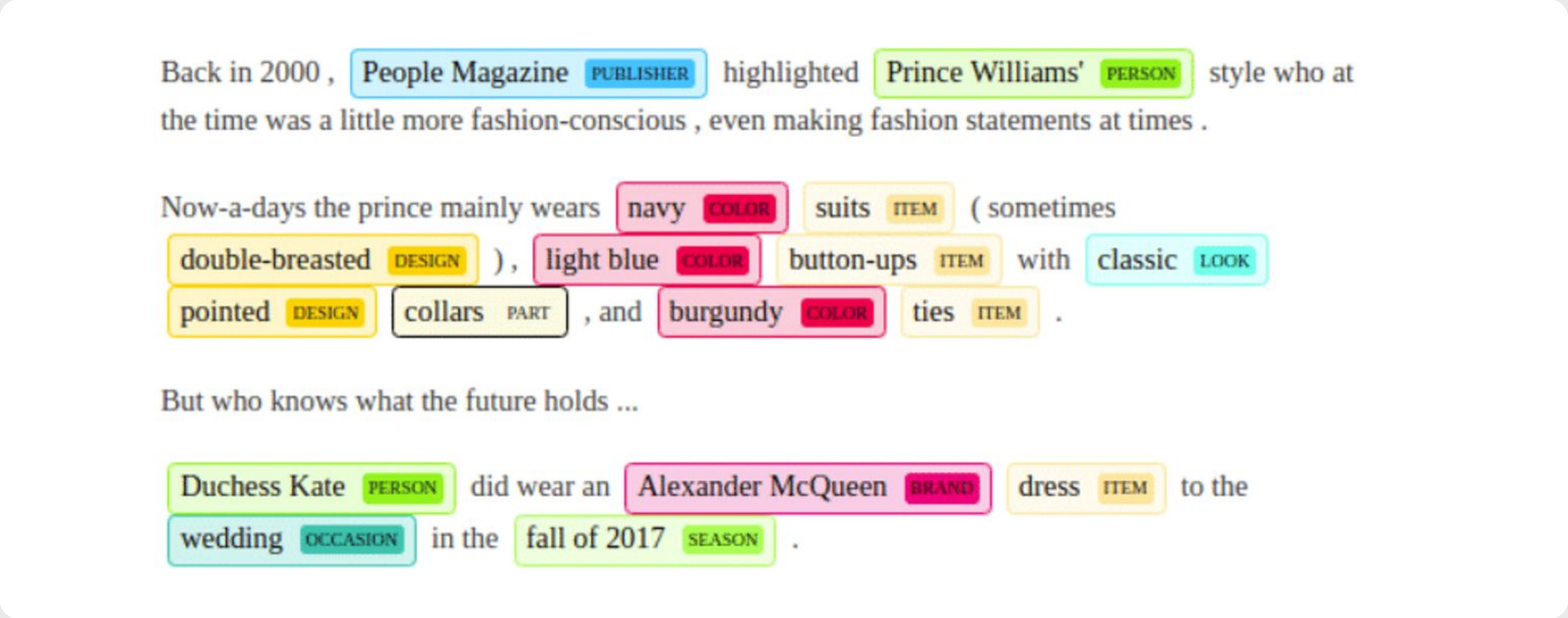
Text-to-Speech Synthesis

Natural Language Generation

World actions / API calls

Slot-filling is information/entity extraction from text

Supervised learning of spans and types. Many tools and base models available. Use LLMs to prototype.



Back in 2000 , People Magazine PUBLISHER highlighted Prince Williams' PERSON style who at the time was a little more fashion-conscious , even making fashion statements at times .

Now-a-days the prince mainly wears navy COLOR suits ITEM (sometimes double-breasted DESIGN) , light blue COLOR button-ups ITEM with classic LOOK pointed DESIGN collars PART , and burgundy COLOR ties ITEM .

But who knows what the future holds ...

Duchess Kate PERSON did wear an Alexander McQueen BRAND dress ITEM to the wedding OCCASION in the fall of 2017 SEASON .

NLU: identifying slot values and domain+intent of request

- Domain classification
    - Asking weather? Booking a flight? Programming alarm clock?
- Intent Determination
    - Find a Movie, Show Flight, Remove Calendar Appt
- Slot Filling
    - Extract the actual slots and fillers

Natural Language Understanding for Filling Slots

- “Show me morning flights from Boston to SF on Tuesday”

```
DOMAIN:      AIR-TRAVEL
INTENT:      SHOW-FLIGHTS
ORIGIN-CITY: “Boston”
ORIGIN-DATE: “Tuesday”
ORIGIN-TIME: “Morning”
DEST-CITY:   “San Francisco”
```

Natural Language Understanding for Filling Slots

- “Turn on my alarm for 6am on May 28”

```
DOMAIN:      ALARM-CLOCK
INTENT:      SET-ALARM
TIME: 2024-05-28 0600
```

Rule-based Slot-Filling

- Write regular expressions or grammar rules

```
Wake me (up) | set (the|an) alarm | get me up
```

- Do text normalization
- Time consuming and brittle NLU capabilities
- With modern NLP tools/features, only use rules alone in special cases
- Simple rules + LLM few-shot recognizers might be just as easy and more robust

Generation Component (NLG)

- **Content Planner:** decides what content to express to user
  - (ask a question, present an answer, etc)
  - Often merged with dialogue manager
- **Language Generation:** chooses syntax and words
  - TTS produces audio
- **In practice:** template-based w/most words prespecified:

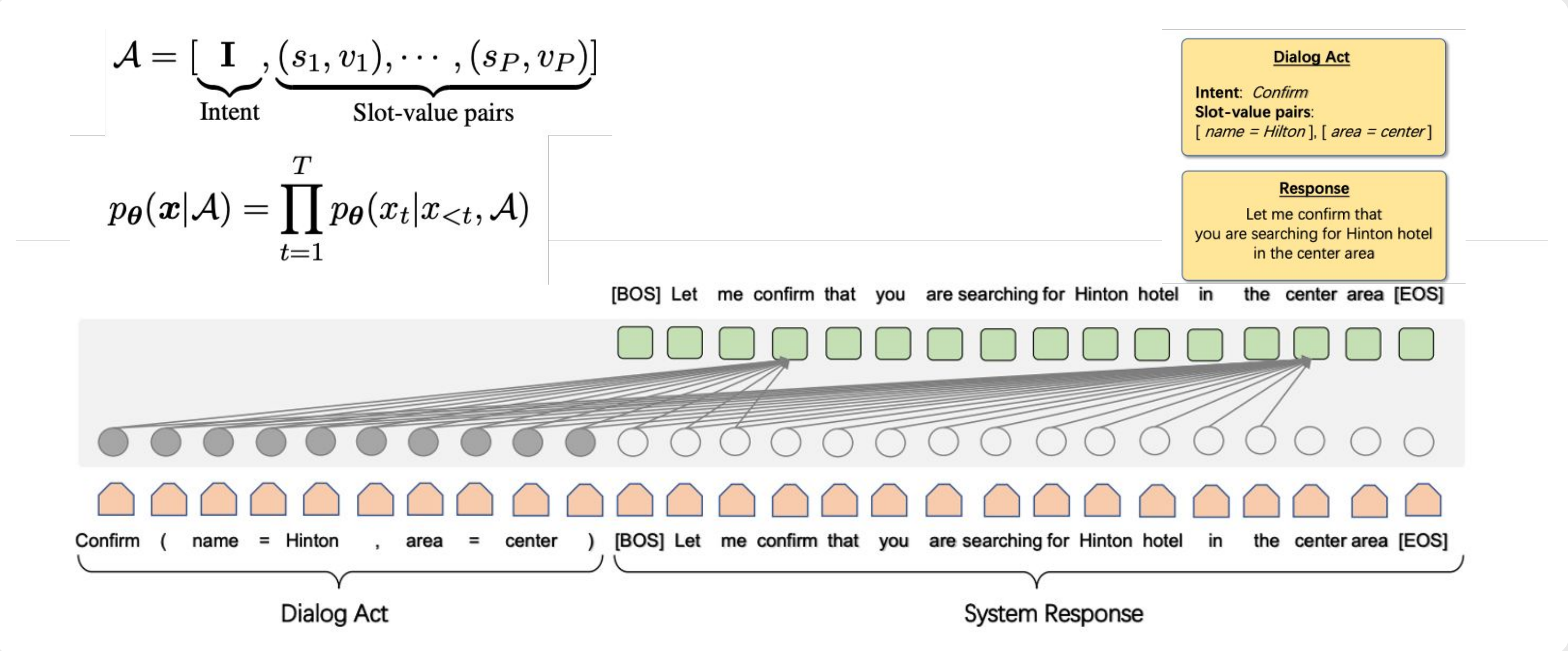
```
What time do you want to leave CITY-ORIG?

Will you return to CITY-ORIG from CITY-DEST?
```

More Sophisticated NLG

- Dialogue manager builds representation of meaning of utterance to be expressed
- Passes this to a “generator”. Old style was templates, modern systems use LLMs
- LLM-based NLG constrained to convey dialog representations can improve user satisfaction

- Critical aspect: Ensure correctness of what we convey to the user!

Semantically Conditioned GPT for Dialog NLG



$$\mathcal{A} = [\underbrace{\mathbf{I}}_{\text{Intent}}, \underbrace{(s_1, v_1), \cdots, (s_P, v_P)}_{\text{Slot-value pairs}}]$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathcal{A}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(x_t|x_{<t}, \mathcal{A})$$

Dialog Act

Intent: *Confirm*
Slot-value pairs:
[*name = Hilton*], [*area = center*]

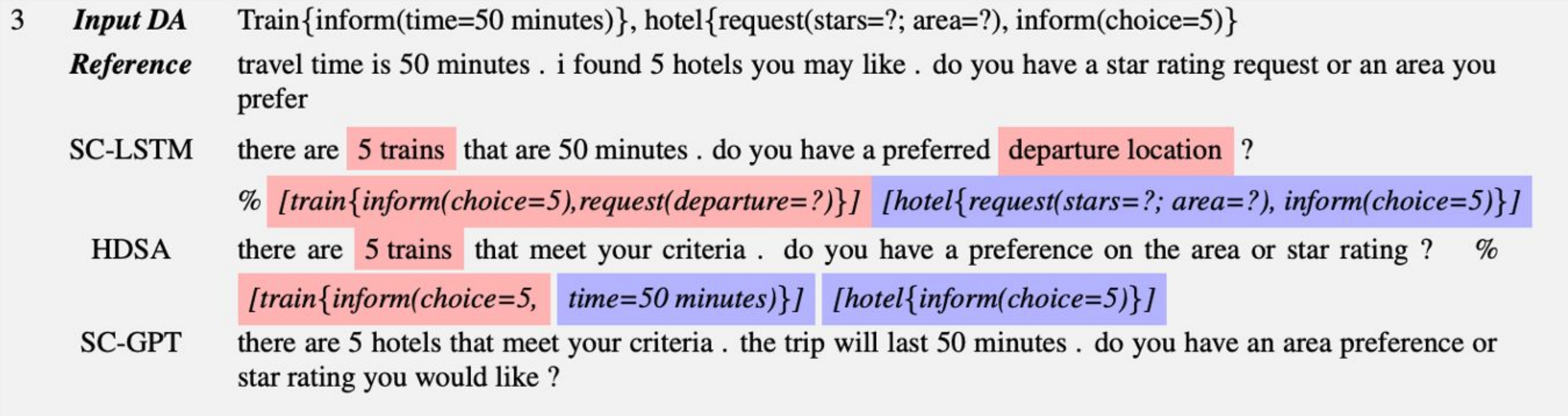
Response

Let me confirm that
you are searching for Hinton hotel
in the center area

[BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Confirm (name = Hinton , area = center) [BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Dialog Act
System Response

Figure: Illustration of SC-GPT. In this example, SC-GPT generates a new word token (e.g. “confirm” or “center”) by attending the entire dialogue act and word tokens on the left within the response. (Peng *et al*, 2020)

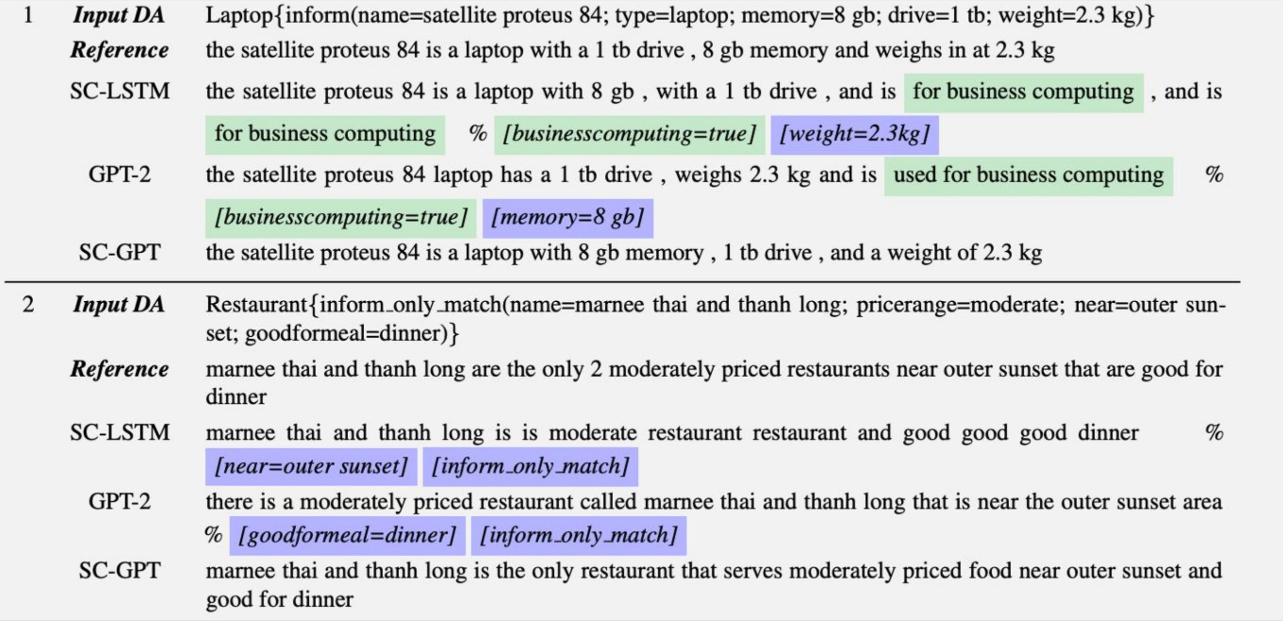
Deep Learning NLG Conditioned on Dialog Semantics



3	***Input DA***	Train{inform(time=50 minutes)}, hotel{request(stars=?; area=?), inform(choice=5)}
	Reference	travel time is 50 minutes . i found 5 hotels you may like . do you have a star rating request or an area you prefer
	SC-LSTM	there are 5 trains that are 50 minutes . do you have a preferred departure location ? % *[train{inform(choice=5),request(departure=?)}]* *[hotel{request(stars=?; area=?), inform(choice=5)}]*
	HDSA	there are 5 trains that meet your criteria . do you have a preference on the area or star rating ? % *[train{inform(choice=5,* *time=50 minutes)}]* *[hotel{inform(choice=5)}]*
	SC-GPT	there are 5 hotels that meet your criteria . the trip will last 50 minutes . do you have an area preference or star rating you would like ?

Figure: Examples of generated utterances from different models along with its corresponding dialogue acts (DAs) and references. Sampled from MultiWOZ. Each generated utterance is followed by a brief description explaining the errors (starting with %). Better viewed in color wrong, redundant, missing information. (Peng *et al*, 2020)

Deep Learning NLG Conditioned on Dialog Semantics



1	***Input DA***	Laptop{inform(name=satellite proteus 84; type=laptop; memory=8 gb; drive=1 tb; weight=2.3 kg)}
	Reference	the satellite proteus 84 is a laptop with a 1 tb drive , 8 gb memory and weighs in at 2.3 kg
	SC-LSTM	the satellite proteus 84 is a laptop with 8 gb , with a 1 tb drive , and is for business computing , and is for business computing % *[businesscomputing=true]* *[weight=2.3kg]*
	GPT-2	the satellite proteus 84 laptop has a 1 tb drive , weighs 2.3 kg and is used for business computing % *[businesscomputing=true]* *[memory=8 gb]*
	SC-GPT	the satellite proteus 84 is a laptop with 8 gb memory , 1 tb drive , and a weight of 2.3 kg
2	***Input DA***	Restaurant{inform_only_match(name=marnee thai and thanh long; pricerange=moderate; near=outer sunset; goodformeal=dinner)}
	Reference	marnee thai and thanh long are the only 2 moderately priced restaurants near outer sunset that are good for dinner
	SC-LSTM	marnee thai and thanh long is is moderate restaurant restaurant and good good good dinner % *[near=outer sunset]* *[inform_only_match]*
	GPT-2	there is a moderately priced restaurant called marnee thai and thanh long that is near the outer sunset area % *[goodformeal=dinner]* *[inform_only_match]*
	SC-GPT	marnee thai and thanh long is the only restaurant that serves moderately priced food near outer sunset and good for dinner

Figure: Examples of generated utterances from different models along with its corresponding dialogue acts (DAs) and references. The examples are sampled from FewShotWOZ. Each generated utterance is followed by a brief description explaining the errors (starting with %). Better viewed in color wrong, redundant, missing information. (Peng *et al*, 2020)

Example dialog system dev frameworks

Dialog frameworks and trade-offs as a developer

- Best experience for user, best voice, etc. may not be easiest to develop, debug, and maintain

- *Developer ergonomics:* designing and optimizing how code frameworks, APIs, and tools impact the efficiency, ease of use, and maintainability of software systems for engineers

- Questions for dialog frameworks as a developer:
    - How do I declare the actions / API calls to connect my dialog manager to taking actions?
        - Is there a pre-defined API format?
        - Do I need to build separate APIs to interface with the dialog system?

    - How do I define conversational behavior and how I want utterances to map to actions / slot values?

    - Does the framework force certain conventions around declaring interactions or action calls?

    - Where can I use this framework (web, mobile, home devices)?

    - Does it support common actions and systems I need? (e.g. outbound calling, emails, DB integration)

Dialog framework examples

- **Alexa skills kit**
  - Clear format for what a “skill” is and explicit slots & types supported.
  - Restrictive interaction model compared to super flexible LLM-based systems but reliable for frame-based dialog
  - Robust simulators, training help, customization of NLU/NLG. Easy development cycles
  - Limited set of endpoints / devices supported. Amazon investing less in smart speakers for Alexa

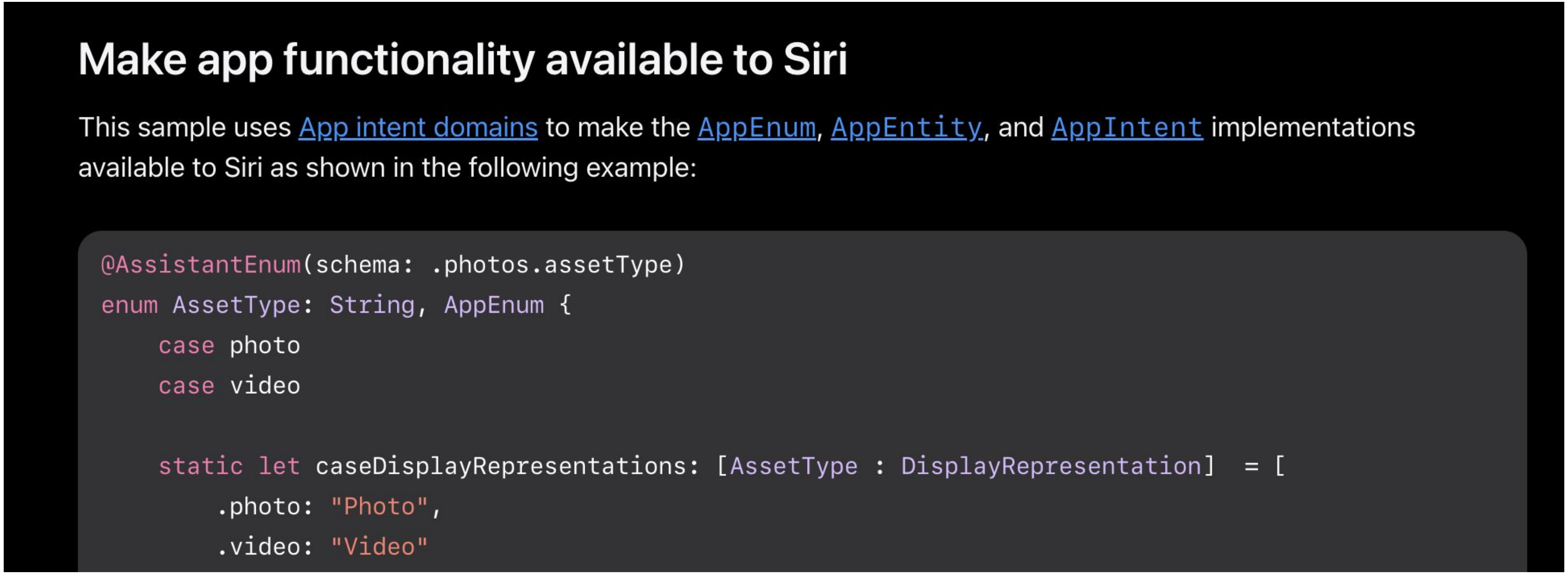
- **Apple/Siri App Intents**
  - Ask developers to build separate set of APIs to connect app functions. Possibly hard to maintain
  - Register app intents using class structure + decorators to implement required hooks for Siri
  - No interaction development required! Siri should “just work” to use app functions supplied to it
    - Downside of this: Unclear what language interaction behavior to expect or how to shape interaction flow.

- **Gridspace dialog builder**
  - LLM-based with separate modules for common call center agent workflow needs: Knowledge base for Q&A, design task flows via playbooks (e.g. password reset, appointment scheduling)
  - System learns/adapts based on sample session data (coaching sessions). Interaction changes the system!
  - Increasing support for complex actions (e.g. email+sales databases, outbound calling to gather info)

Siri App Intents

Code example for declaring app actions in a way that is available to Siri assistant to use (docs)

No explicit interaction design required. No simulator available for Siri + your app intent set, can’t test yet.



Make app functionality available to Siri

This sample uses App intent domains to make the `AppEnum`, `AppEntity`, and `AppIntent` implementations available to Siri as shown in the following example:

```
@AssistantEnum(schema: .photos.assetType)
enum AssetType: String, AppEnum {
    case photo
    case video

    static let caseDisplayRepresentations: [AssetType : DisplayRepresentation]  = [
        .photo: "Photo",
        .video: "Video"
```

Alexa Skills Kit

- **A Skill is a top level command for Alexa**
    - “Alexa open 224S Homework 2”
    - Skill **-> domain ontology**

- **A skill contains intents which are distinct task actions**
    - Intent **-> frame**
    - Design intents with built-in capabilities per intent and ASK interaction model in mind

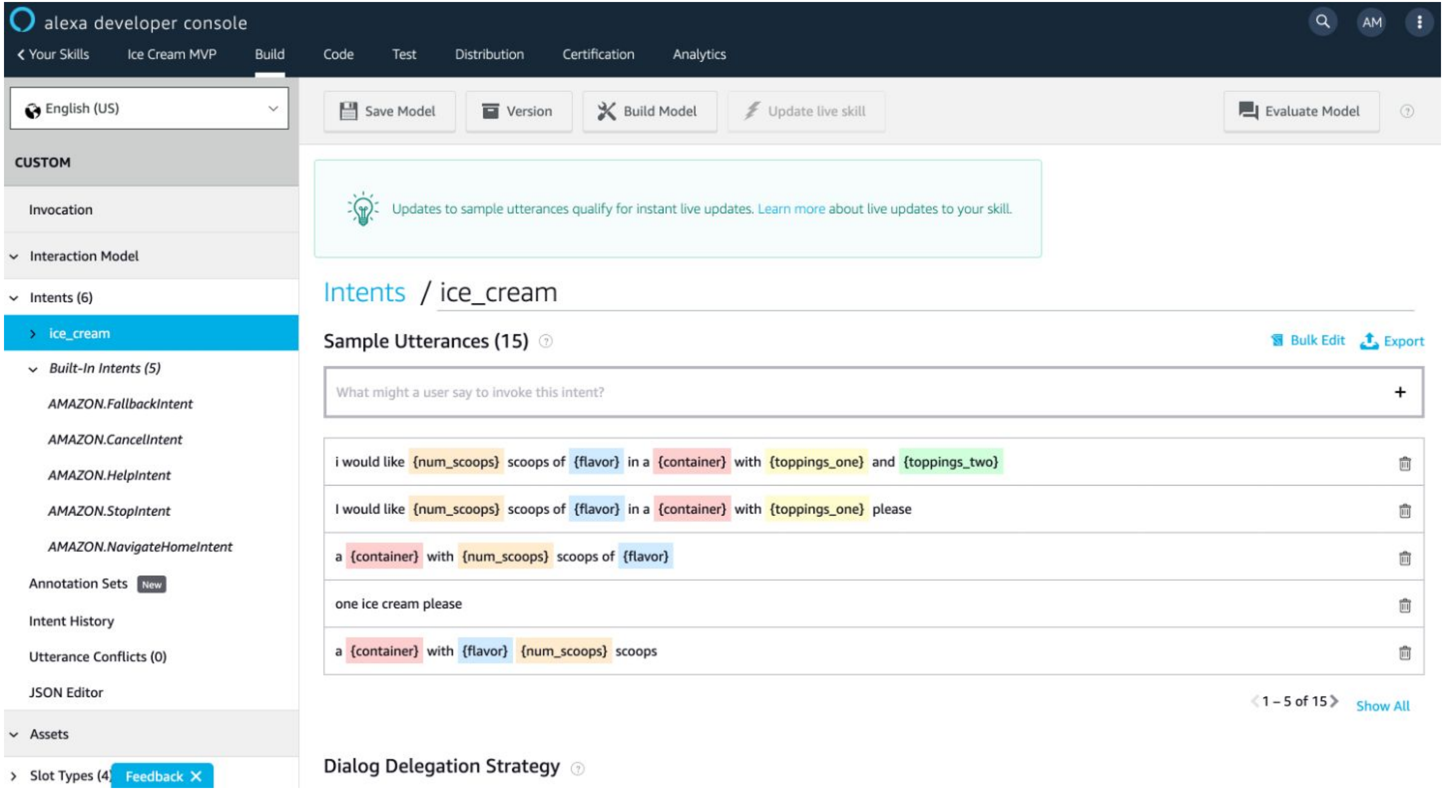
- **Each intent contains slots which each have a slot type and take on a slot value**

- **Not quite this simple (e.g. ASK built-in intents are not simple to define in the frame/slot abstraction)**

Alexa Skills Kit

- Dialog management is complex, partially handled with built-in features (clarification, value verification, cancel skill, etc)
- NLU through grammars and examples.
  - ASK trains models for you based on examples
  - Many rich slot types (dates, numbers, lists)
- Task management is custom! ASK provides a dialogue API to your web server, you implement server-side task execution.
- NLG is template-based with ASK adding variety
- ASR/TTS handled by ASK. Interface is text/transcripts
- Overall framework is API/SDK oriented like web dev

Alexa intent and slot examples (2020)



Updates to sample utterances qualify for instant live updates. Learn more about live updates to your skill.

Intents / ice_cream

Sample Utterances (15)

What might a user say to invoke this intent?

i would like {num_scoops} scoops of {flavor} in a {container} with {toppings_one} and {toppings_two}

I would like {num_scoops} scoops of {flavor} in a {container} with {toppings_one} please

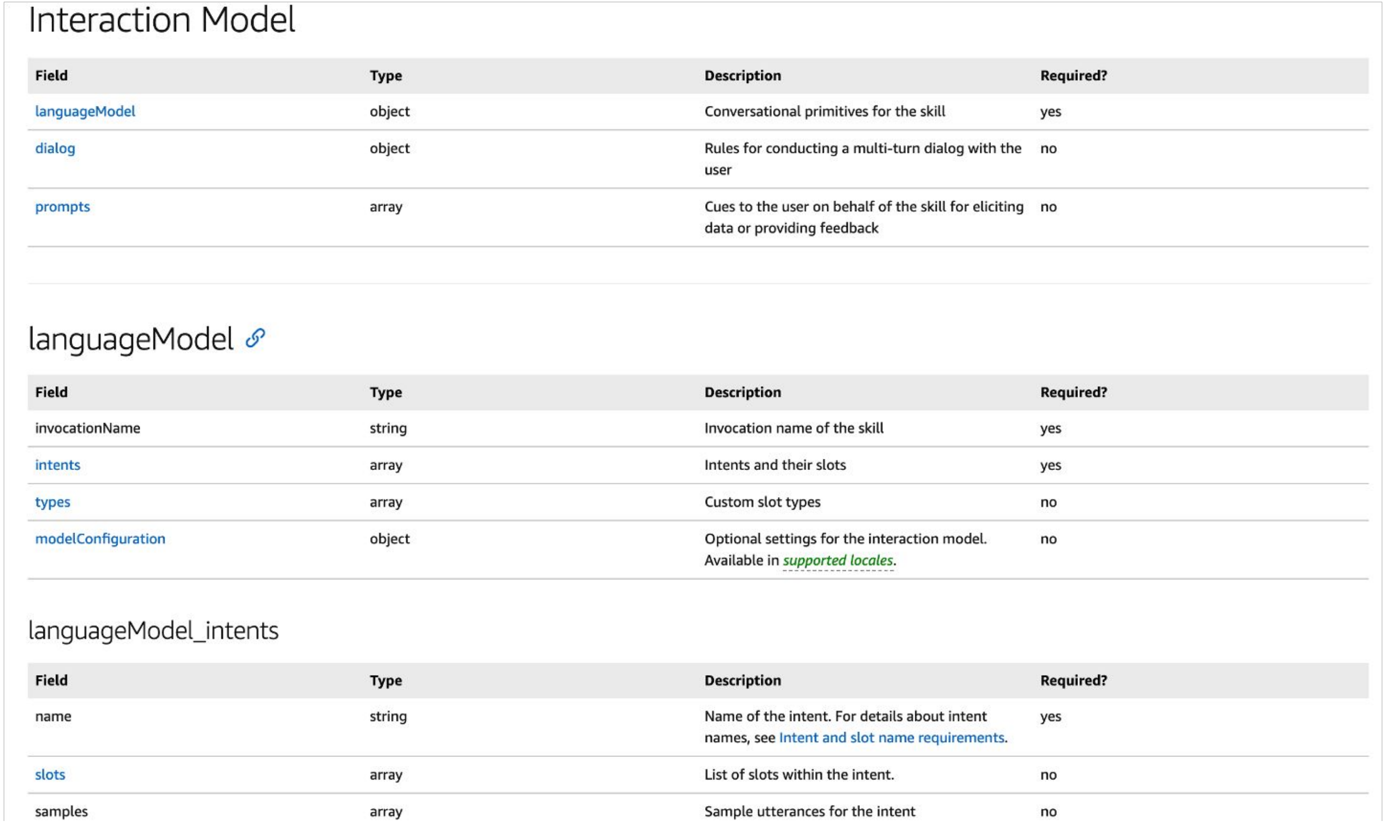
a {container} with {num_scoops} scoops of {flavor}

one ice cream please

a {container} with {flavor} {num_scoops} scoops

Dialog Delegation Strategy

ASK Interaction Schema



Interaction Model

Field	Type	Description	Required?
languageModel	object	Conversational primitives for the skill	yes
dialog	object	Rules for conducting a multi-turn dialog with the user	no
prompts	array	Cues to the user on behalf of the skill for eliciting data or providing feedback	no

languageModel

Field	Type	Description	Required?
invocationName	string	Invocation name of the skill	yes
intents	array	Intents and their slots	yes
types	array	Custom slot types	no
modelConfiguration	object	Optional settings for the interaction model. Available in *supported locales*.	no

languageModel_intents

Field	Type	Description	Required?
name	string	Name of the intent. For details about intent names, see Intent and slot name requirements.	yes
slots	array	List of slots within the intent.	no
samples	array	Sample utterances for the intent	no

(ASK docs)

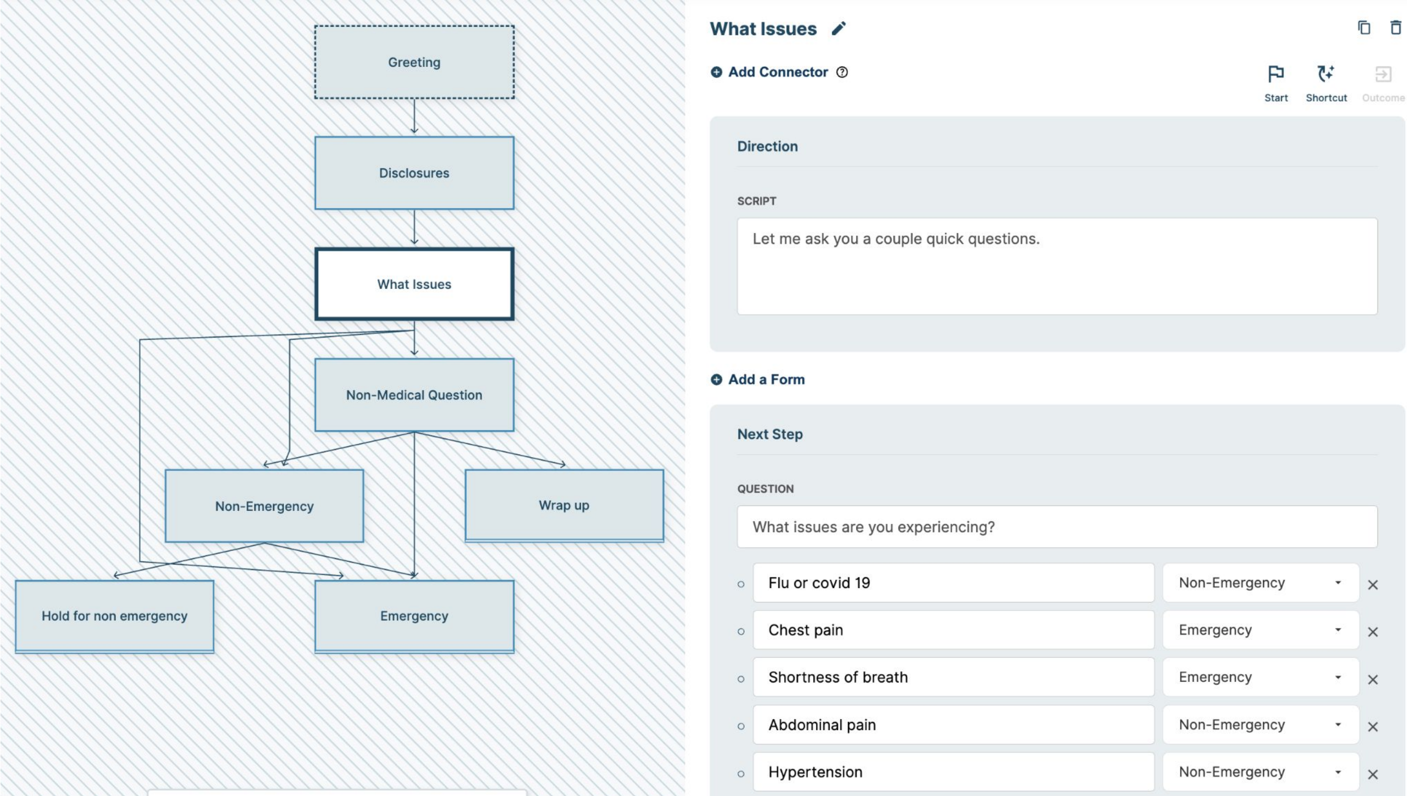
ASK Intent JSON Example

This example shows a portion of the intent object for a `PlanMyTrip` intent. The utterances for the intent are in `interactionModel.languageModel.intents[].samples`. Each slot has its own samples array. For brevity, other properties within `interactionModel` and `languageModel` are not shown

```
{ "interactionModel": { "languageModel": { "intents":
 [ { "name": "PlanMyTrip", "slots": [ { "name": "travelDate", "type": "AMAZON.DATE",
  "samples": [ "I am taking this trip on {travelDate}", "on {travelDate}", "{travelDate}" ] },
{ "name": "toCity", "type": "AMAZON.US_CITY", "samples": [ "I'm going to {toCity}", "{toCity}" ] },
{ "name": "fromCity", "type": "AMAZON.US_CITY", "samples": [ "{fromCity}", "I'm starting from {fromCity}" ] },
{ "name": "travelMode", "type": "LIST_OF_TRAVEL_MODES", "samples": [ "I am going to {travelMode}", "{travelMode}" ] },
{ "name": "activity", "type": "LIST_OF_ACTIVITIES", "samples": [ "{activity}", "I plan to {activity}" ] } ],
 "samples": [ "{toCity}", "I want to travel from {fromCity} to {toCity} {travelDate}", "i want to visit {toCity}", "i am going
on trip on {travelDate}", "I'm {travelMode} from {fromCity} to {toCity}", "i'm {travelMode} to {toCity} to {activity}", "plan a
trip", "plan a trip to {toCity} ", "plan a trip starting from {fromCity} ", "I'd like to leave on {travelDate} ", "I'd like to
leave on the {travelDate} ", "I'd like to fly out of {fromCity} " ] } ] } }
```

(ASK docs)

Gridspace dialog builder



SCRIPT

Let me ask you a couple quick questions.

QUESTION

What issues are you experiencing?

Flu or covid 19	Non-Emergency
Chest pain	Emergency
Shortness of breath	Emergency
Abdominal pain	Non-Emergency
Hypertension	Non-Emergency

See live demos on Gridspace.com

Questions?

Alexa Domain Classification

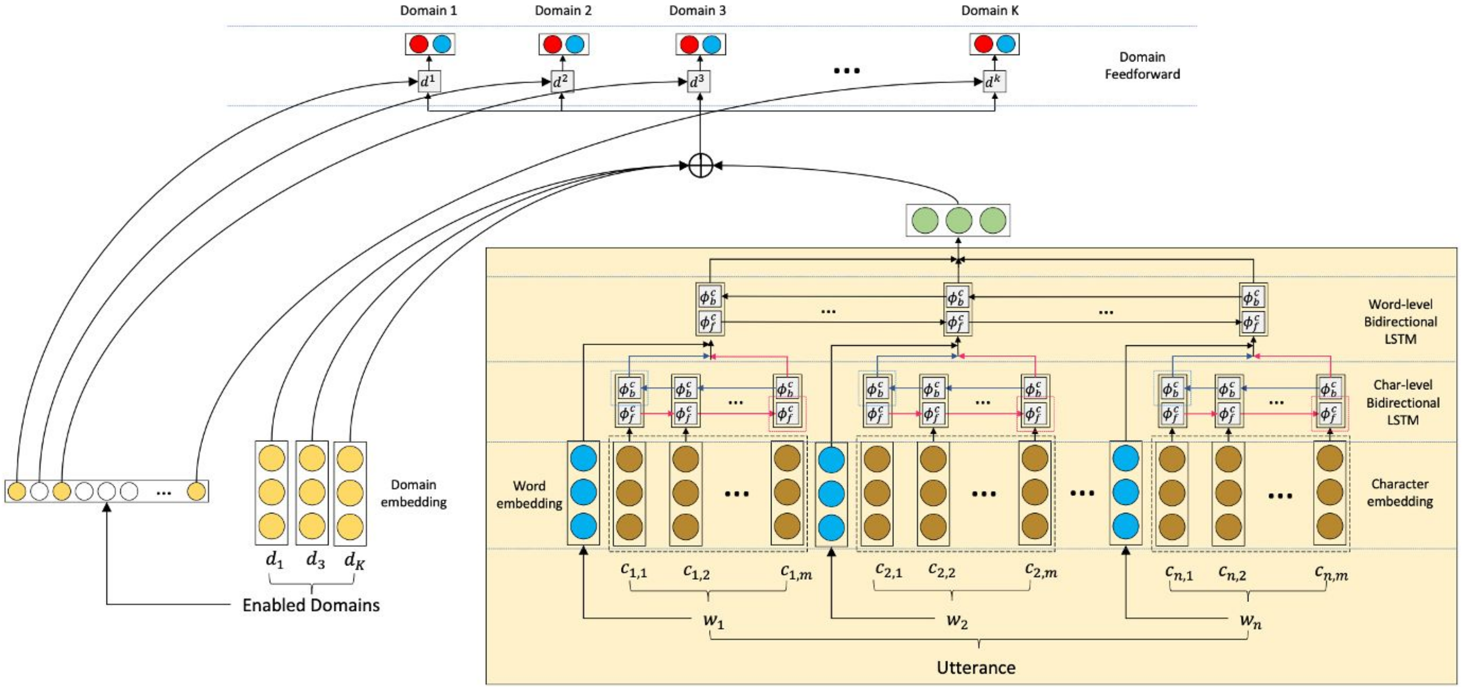
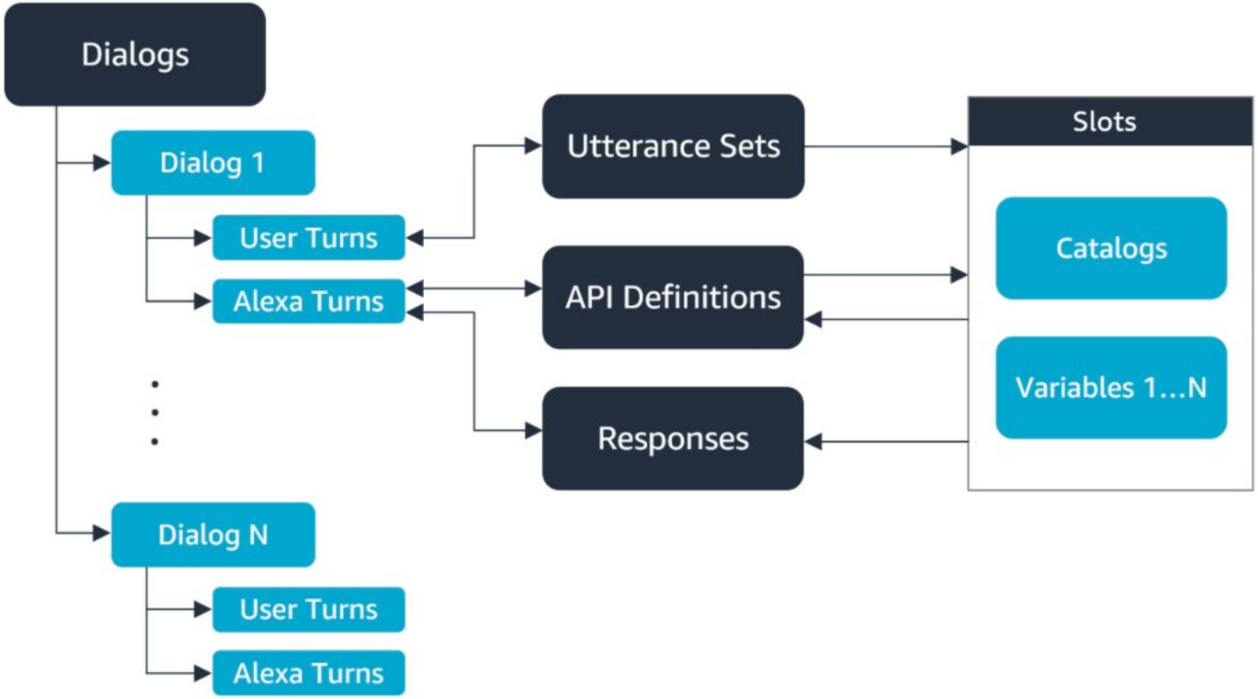


Figure: The overall architecture of personalized dynamic domain classifier. Kim et al, 2018

Alexa Conversations (2020)

When you build an Alexa Conversation skill, you create the following components that train Alexa Conversation how to interact with your user.



Announcement, ASK docs