Using Language Technologies Spoken Dialog Systems



EECS 183/283a: Natural Language Processing

Applications of Spoken Dialog



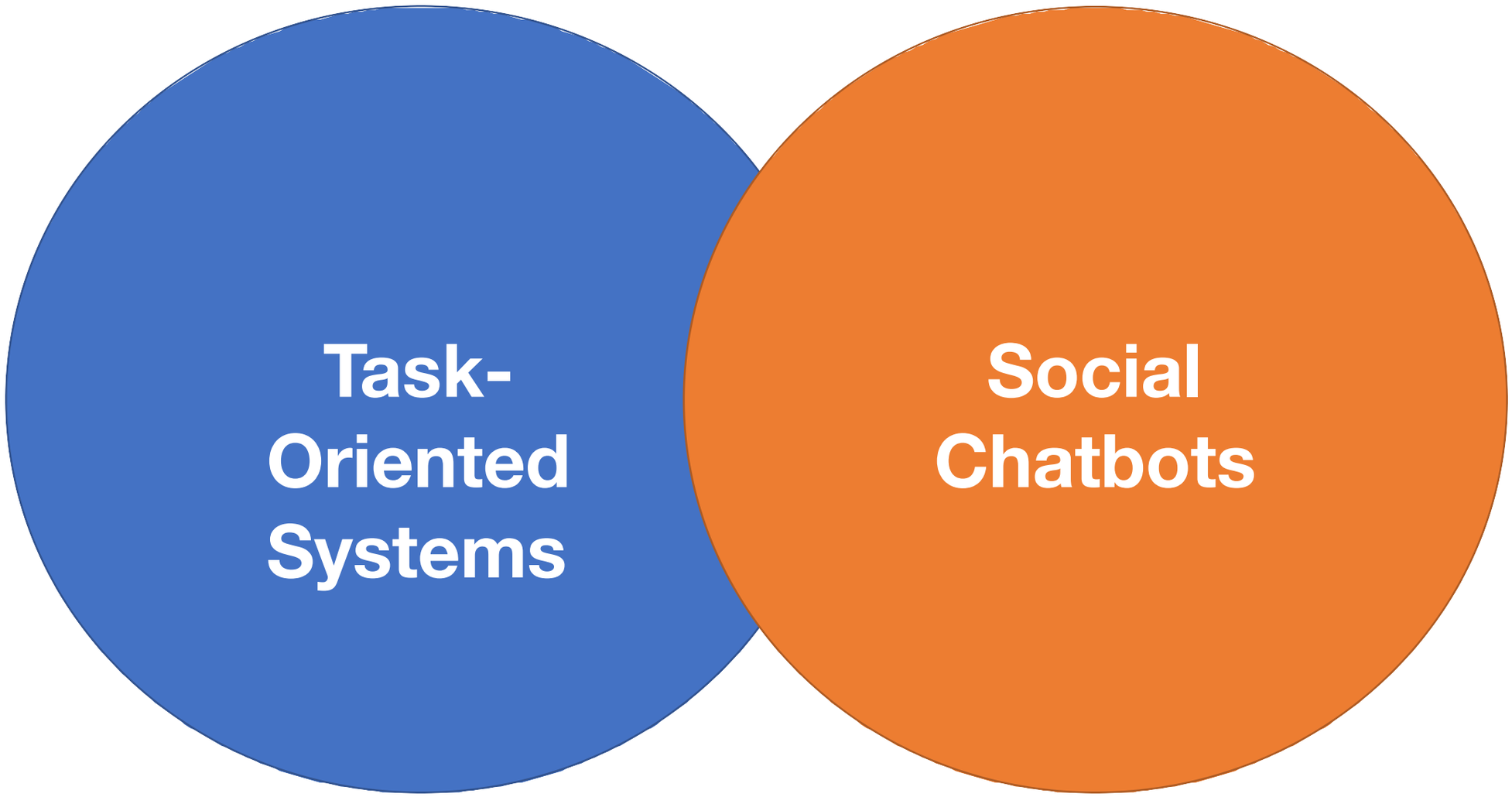
Applicable in various areas, such as and entertainment, education and health care.

- Entertainment: Create targeted advertisement (Yu et al., IJCAI 2017, SLT 2016) Social chatbots (Amazon Alexa Prize, Cohn et al, SIGDIAL 2019)
- Education: Provide training (Yu et al., IWSDS 2016) and facilitate discussion on MOOCs
- Health care: Support therapy for depression (Yu et al., SEMDIAL 2013), aphasia and dementia, persuade people to perform physical exercise
- Customer Service: Provide information and perform actions (Shi&Yu ACL 2018, Zhang&Yu SIGDIAL 2018, Qian&Yu ACL 2019)
- Marketing: Persuade people to donate to charity (Wang et al., ACL 2019 best paper nomination), Anti-Scam (DARPAASED)

Applicable in various platforms: virtual agents and robotics

- Virtual: Build characters for video games
- Robots: Service robots, e.g. direction giving (Yu et al., SIGDIAL 2015), nursing and rescuing

Types of Dialog Systems



Task-Oriented Systems



Users and systems have aligned goals.



Flight booking



Restaurant booking



Bus Information

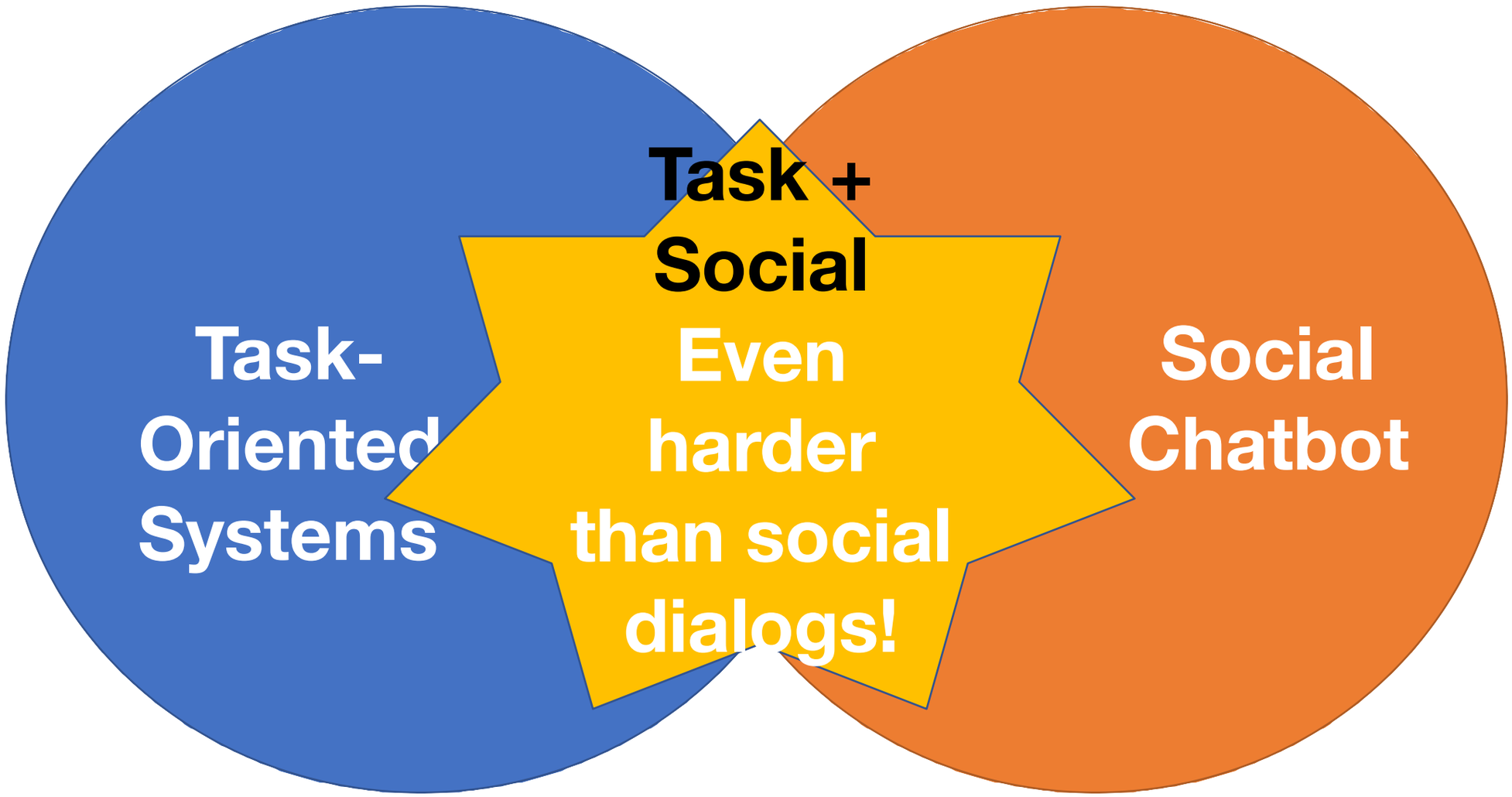
Social Chatbots

Talk about anything with anyone.





Dialog System Types



Outline



- Human conversation
  - Grounding
  - Dialog acts
- Dialog systems
  - Conceptual architecture
  - Dialog manager
  - Initiative

Task-Oriented Human Conversation



- Turn-taking
- Speech Acts
- Grounding

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 26.1 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.

Turn-taking

Dialogue is characterized by turn-taking.

A:

B:

A:

B:

…

So how do speakers know when to take the floor?

Adjacency pairs

Sacks et al. (1974)

- **Adjacency pairs**: current speaker selects next speaker
  - Question/answer
  - Greeting/greeting
  - Compliment/downplayer
  - Request/grant

- Silence inside the pair is meaningful:

A: Is there something bothering you or not?
(1.0)
A: Yes or
no? (1.5)
A: Eh
B: No.

Speech Acts



- Austin (1962): An utterance is a kind of action
- Clear case: performatives
- I name this ship the Titanic
- I second that motion
- I bet you 5 dollars it will snow
- Performative verbs (name, second)
- Locutionary (what was said)
- Illocutionary (what was meant)

5 classes of "speech acts"

Searle (1975)



Assertives: committing the speaker to something's being the case

(suggesting, putting forward, swearing, boasting, concluding)

Directives: attempts by speaker to get addressee to do something

(asking, ordering, requesting, inviting, advising, begging)

Commissives: Committing speaker to future course of action

(promising, planning, vowing, betting, opposing)

Expressives: expressing psychological state of the speaker about a state of affairs

(thanking, apologizing, welcoming, deploring).

Declarations: changing the world via the utterance

(I resign; You're fired)

More Illocutionary acts: Grounding

- Why do elevator buttons light up?

- Clark (1996) (after Norman 1988)

  Principle of closure. Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it

- What is the linguistic correlate of this?

Grounding



- Need to know whether an action succeeded or failed
- Dialogue is also an action
  - a collective action performed by speaker and hearer
    - Common ground: set of things mutually believed by both speaker and hearer
- Need to achieve common ground, so hearer must ground or acknowledge speakers utterance.

How do speakers ground? Clark and Schaefer

- Continued attention:
  - B continues attending to A
- Relevant next contribution:
  - B starts in on next relevant contribution
- Acknowledgement:
  - B nods or says continuer (uh-huh) or assessment (great!)
- Demonstration:
  - B demonstrates understanding A by **reformulating** A's contribution, or by **collaboratively completing** A's utterance
- Display:
  - B repeats verbatim all or part of A's presentation

A human-human conversation

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.

Grounding examples

Display:

C: I need to travel in May

A: And, what day in May did you want to travel?

Acknowledgement

C: I want to fly from Boston

A: mm-hmm

C: to Baltimore Washington International

Grounding Examples (2)

- Acknowledgement + next relevant contribution
- And, what day in May did you want to travel?
- And you're flying into what city?
- And what time would you like to leave?

- The and indicates to the client that agent has successfully understood answer to the last question.

Grounding negative responses
From Cohen et al. (2004)

- System: Did you want to review some more of your personal profile?
- Caller: No.
- System: Okay, what's next?

Good!

- System: Did you want to review some more of your personal profile?
- Caller: No.
- System: What's next?

Bad!

Generalized idea -- Dialog acts



- Also called “conversational moves”
- An act with (internal) structure related specifically to its dialogue function
- Incorporates ideas of grounding
- Varies by task, domain, etc. but core concepts like grounding often present

Explicit confirmation

U: I'd like to fly from Denver Colorado to New York City on September 21st in the morning on United Airlines

S: Let's see then. I have you going from Denver Colorado to New York on September 21st. Is that correct?

U: Yes

Implicit confirmation: display



U: I'd like to travel to Berlin

S: When do you want to travel to Berlin?

U: Hi I'd like to fly to Seattle Tuesday morning

S: Traveling to Seattle on Tuesday, August eleventh in the morning. Your name?

Grice's Maxims of Conversation

- **The maxim of quantity**,
  - where one tries to be as informative as one possibly can, and gives as much information as is needed, and no more.

- **The maxim of quality**,
  - where one tries to be truthful, and does not give information that is false or that is not supported by evidence.

- **The maxim of relation**,
  - where one tries to be relevant, and says things that are pertinent to the discussion.

- **The maxim of manner**,
  - when one tries to be as clear, as brief, and as orderly as one can in what one says, and where one avoids obscurity and ambiguity.

Verbmobil task

- Two-party scheduling dialogues
- Speakers were asked to plan a meeting at some future date
- Data used to design conversational agents which would help with this task
- (cross-language, translating, scheduling assistant)

Verbmobil Dialogue Acts



THANK	thanks
GREET	Hello Dan
INTRODUCE	It's me again
BYE	Allright, bye
REQUEST-COMMENT	How does that look?
SUGGEST	June 13th through 17th
REJECT	No, Friday I'm booked all day
ACCEPT	Saturday sounds fine
REQUEST-SUGGEST	What is a good day of the week for you?
INIT	I wanted to make an appointment with you
GIVE_REASON	Because I have meetings all afternoon Okay
FEEDBACK	Let me check my calendar here
DELIBERATE	Okay, that would be wonderful
CONFIRM	Okay, do you mean Tuesday the 23rd?
CLARIFY	

Two classes of systems



1. Chatbots
2. (Goal-based) Dialog agents
   - *SIRI, interfaces to cars, robots,*
   - *booking flights or restaurants*

The word "chatbots" is sometimes used in the popular press for both. We'll use it only for #1.

Conversational/Dialog Agents



- AKA:
  - Spoken Language Systems
  - Dialogue Systems
  - Speech Dialogue Systems
- Applications:
  - Travel arrangements (Amtrak, United airlines)
  - Telephone call routing
  - Tutoring
  - Communicating with robots
  - Anything with limited screen/keyboard

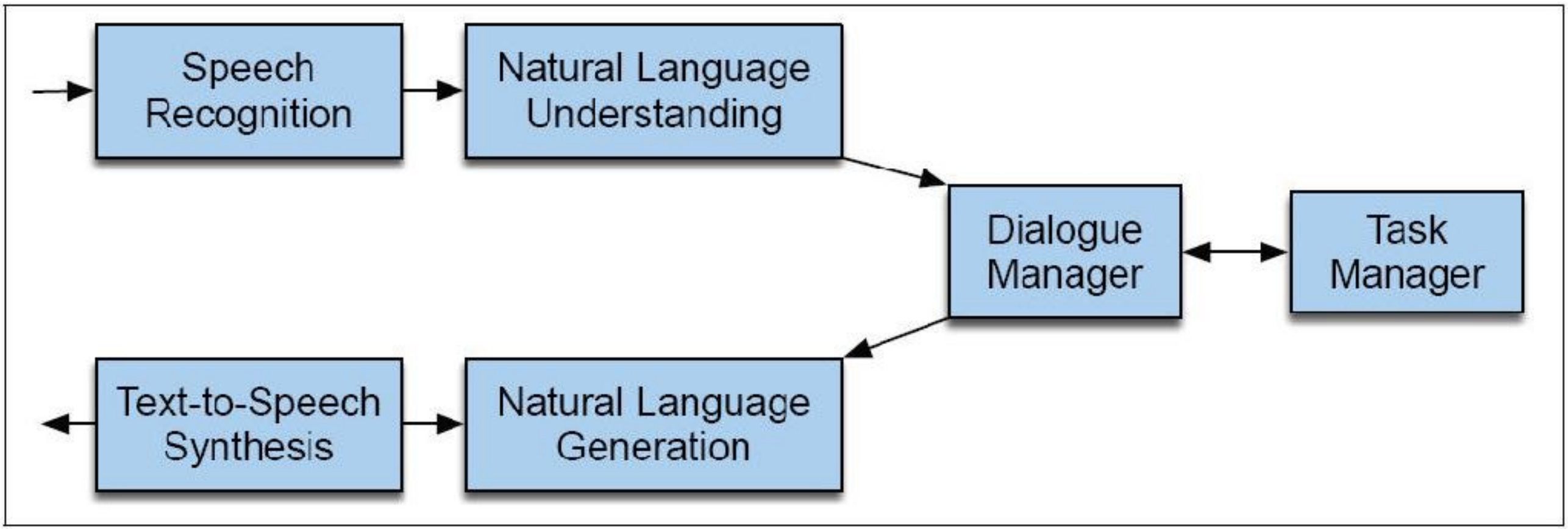
A travel dialog: Communicator Xu and Rudnicky (2000)

S1:	Hello. You've reached the [Communicator]. Tell me your full name
U2:	*hi i'd like to fly to seattle tuesday morning*
S3:	Traveling to Seattle on Tuesday, August eleventh in the morning. Your full name?
U4:	*[John Doe]*
S5:	Hello, [John], I'm glad to hear from you again. You can now ask me for help.
S6:	OK. Let's see I have a United flight departing Pittsburgh at ten forty five a.m. arriving into Seattle at one fifty seven p.m. Is that OK?
U7:	*Yes*
S8:	Will you return to Pittsburgh from Seattle?
U9:	*no i will go to Los Angeles*
S10:	Traveling to Los Angeles. on what date
U11:	*Sorry can we go back to talk about the previous leg again*
S12:	For the first leg, you are on a United flight 1446 departing Pittsburgh at ten forty five A.M. tomorrow, arriving in Seattle at one fifty seven P.M. Do you want to take that?
U13:	*actually i want to leave on wednesday*
S14:	a flight on wednesday.
S15:	I have a flight departing Pittsburgh at ten forty five a.m., arrives Seattle at one fifty seven p.m. Is that OK?
U16:	*Yes*

A tutorial dialogue: ITSPOKE Litman and Silliman (2004)

S: I didn't understand. What force acts on all objects within close proximity to earth?
U: *uh the force of gravity*
S: Fine. Besides the Earth's gravitational force, are there any other forces acting on the person, keys, and elevator after the release?
U: *no*
S: Yep. Thus, keep in mind that after the release, the only force on the person, keys, and elevator is the force of gravity. If gravity is the only force acting on an object, will it be moving or staying still?
U: *it will be moving*
S: Fine. Not only are the person, keys, and elevator moving, they have only gravitational forces on them. When an object is falling and has only gravitational force on it, it is said to be in what?
U: *freefall*

Spoken Dialog Agent Conceptual Architecture



Dialogue Manager



- Controls the architecture and structure of dialogue
  - Takes input from ASR/NLU components
  - Maintains some sort of state
  - Interfaces with Task Manager
  - Passes output to NLG/TTS modules

Dialogue + Task Management



- Often we think of simpler dialog tasks as interactively completing a data structure or **frame**
- Task execution (e.g. making a reservation) can happen via APIs etc.
- Defining the data structure required to complete a task can be difficult and time consuming
- Some modern approaches attempt to learn dialog/task actions directly (e.g. simulate clicks or API calls made by a human agent)

Dialog architecture for Personal Assistants

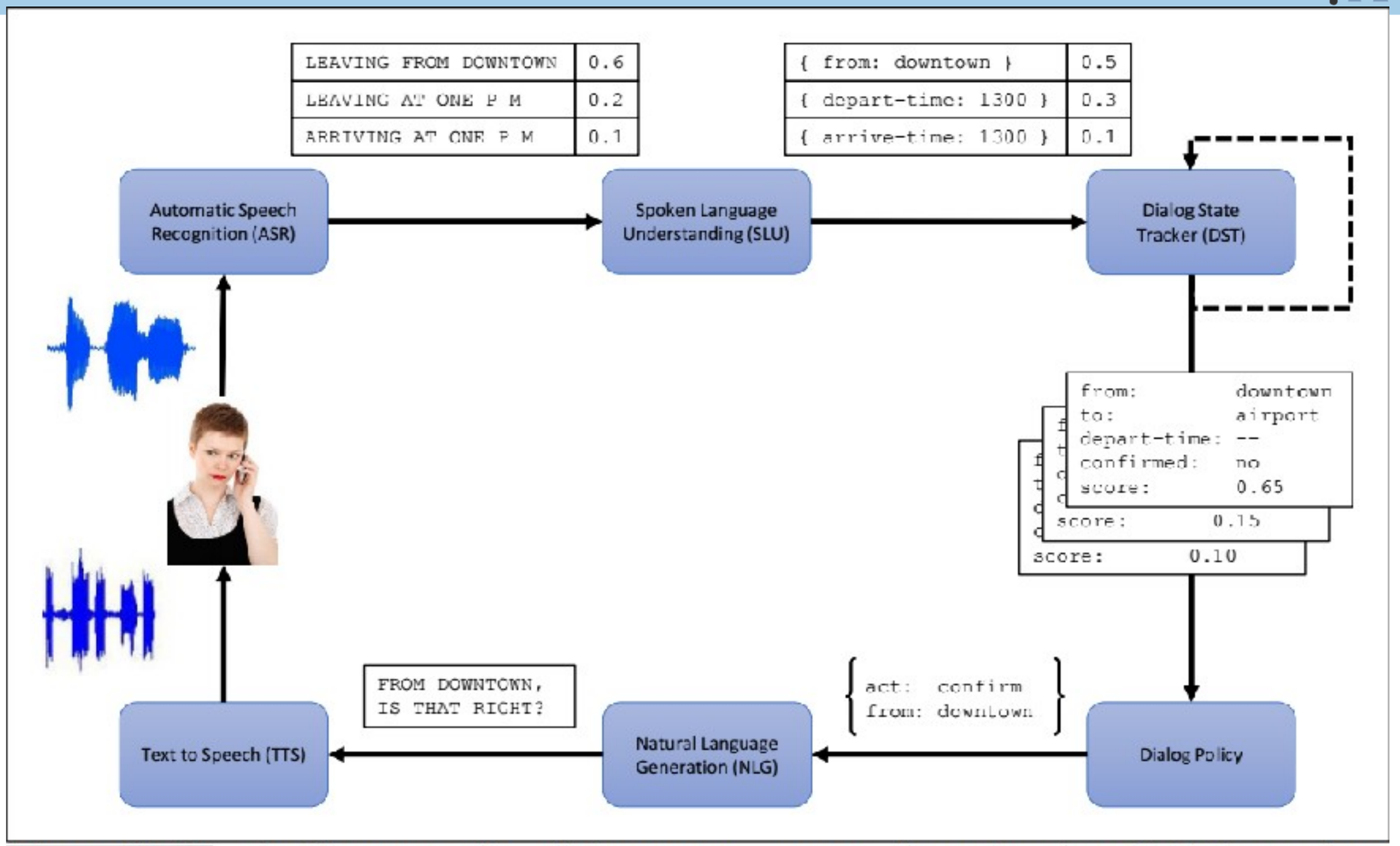


Figure 29.12 Architecture of a dialogue-state system for task-oriented dialogue from (Williams et al., 2016).

Possible architectures for dialog management

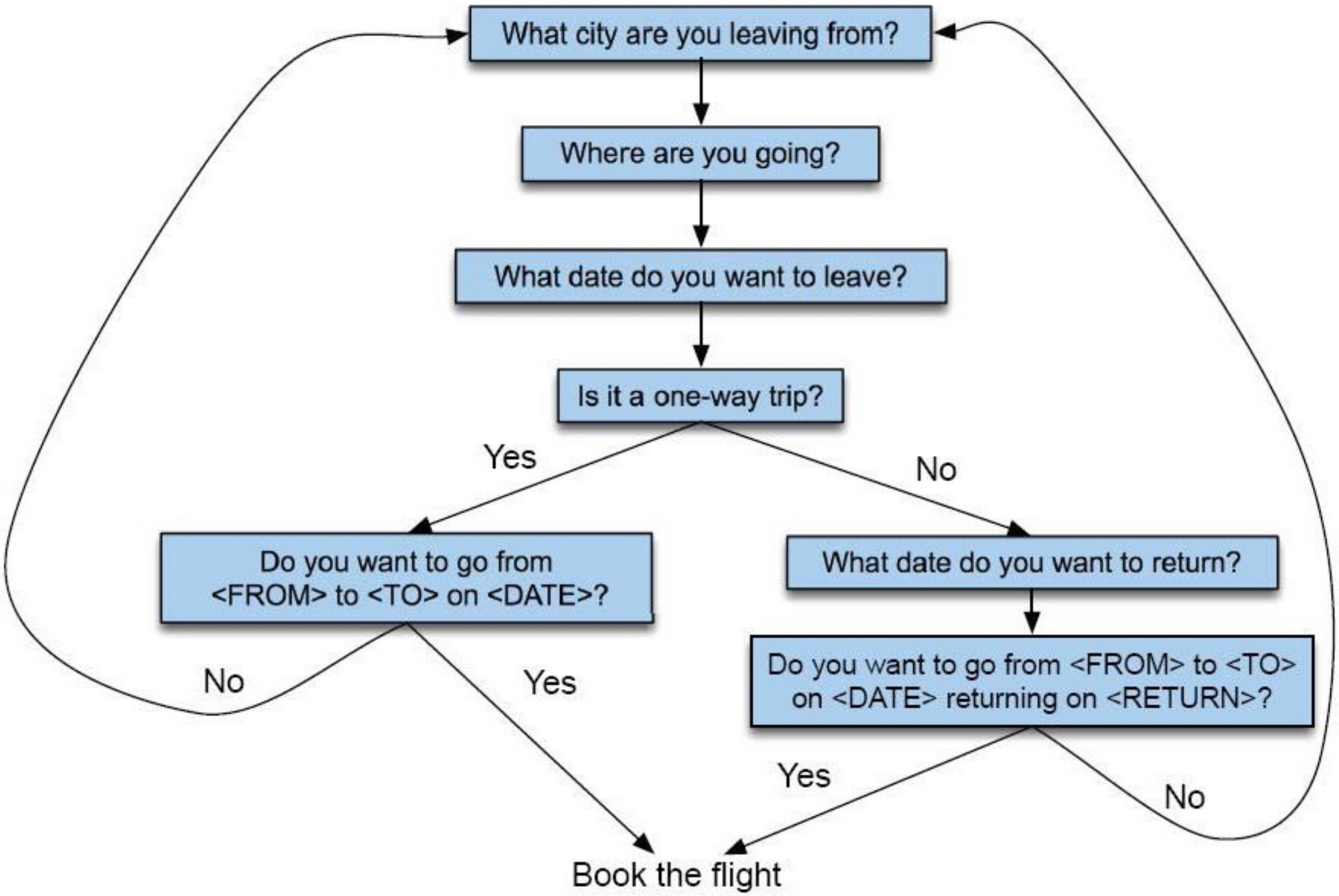


Finite State

Frame-based

Information State (Markov Decision Process)

Distributional / neural network


What city are you leaving from?
Where are you going?
What date do you want to leave?
Is it 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
Yes
Yes
No
Book the flight

Dialogue Initiative



- Systems that control conversation are called **single initiative**.
- **Initiative**: who has control of conversation
- In normal human-human dialogue, initiative shifts back and forth between participants.

User Initiative

- User directs the system
  - Asks a single question, system answers Examples: **Voice web search**
- But system can't:
  - ask questions back,
  - engage in clarification dialogue,
  - engage in confirmation dialogue

Sytem Initiative



System completely controls the conversation

- Simple to build
- User always knows what they can say next
- System always knows what user can say next

+

- Known words: Better performance from ASR
- Known topic: Better performance from NLU
- OK for VERY simple tasks (entering a credit card, or login name and password)

-

- Too limited

Problems with System Initiative

- Real dialogue involves give and take!
- In travel planning, users might want to say something that is not the direct answer to the question.
- For example answering more than one question in a sentence:

Hi, I'd like to fly from Seattle Tuesday morning

I want a flight from Milwaukee to Orlando one way leaving after 5 p.m. on Wednesday.

Single initiative + universals



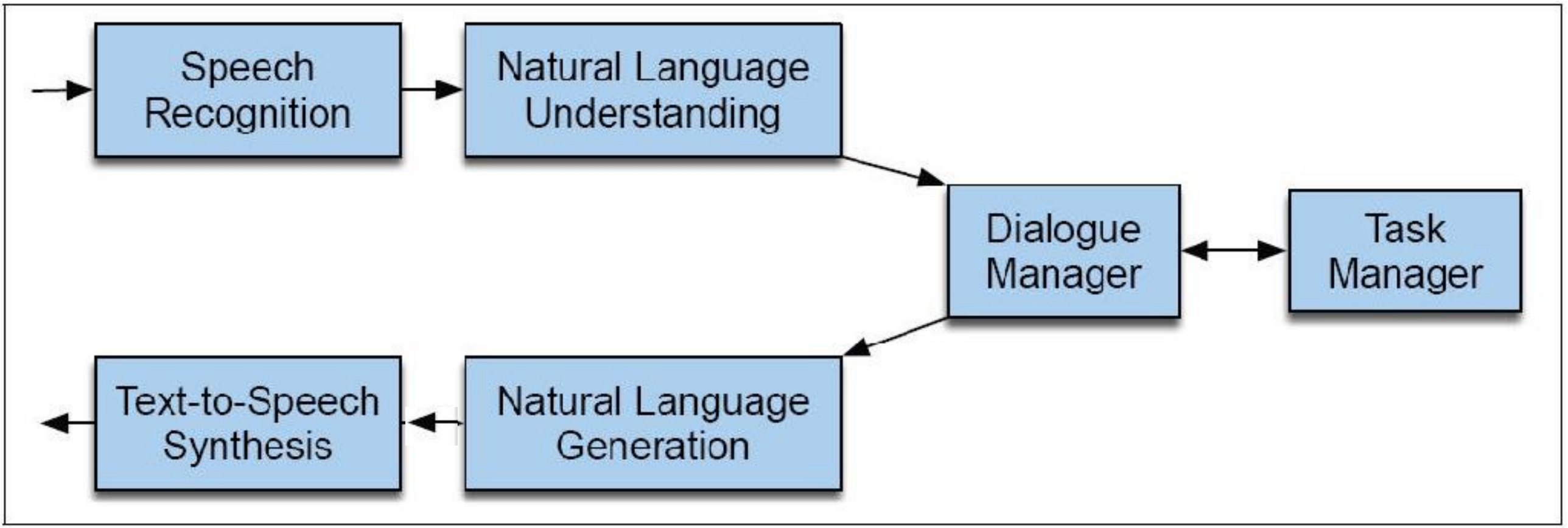
- We can give users a little more flexibility by adding **universals**: commands you can say anywhere
- As if we augmented every state of FSA with these
  Help
  Start over
  Correct
- This describes many implemented systems
- But still doesn’t allow user much flexibility

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

Spoken Dialog Agent Conceptual Architecture



System design considerations



- Goal and scope of overall system?
- What tasks/actions are supported?
  - What state do the dialog/task managers track?
- What level of interaction complexity?
  - Initiative? Back-tracking? NLU support for paraphrasing?
- Need a solution for each module (ASR, TTS, NLU, NLG, task/dialog manger)
- What is the interface / data structure between modules?
  - e.g. Does ASR module send transcripts only? Emotion labels? Audio?

Dialog System Design: User-centered Design

Gould and Lewis 1985

1. Study the user and task
2. Build simulations "Wizard of Oz study"
3. Iteratively test the design on users



Rough system design process



1. Overall system goal.
2. Define set of task actions system can perform
3. Create example interactions
4. Define dialog manager approach (actions + dialog acts/ state of system)
5. Choose NLU approach matching complexity of tasks and approach to initiative + dialog acts
6. Define NLG approach and dialog state -> NLG interface
7. Create a dialog policy (choosing next dialog action and sending to NLG)
8. Choose ASR/TTS approach. Update NLU/NLG if needed

Frame-based dialog agents



- One or more **frames**
  - Each a collection of **slots**
  - Each slot having a **value**
- **Domain ontology** is full collection of frames for a (sub)system
- A frame defines the task data structure

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?

Parsing for Slot filling

- *Parsing of SPEECH not TEXT*
  - *Eh, I wanna go, wanna go to Boston tomorrow*
  - *If its not too much trouble I'd be very grateful if one might be able to aid me in arranging my travel arrangements to Boston, Logan airport, at sometime tomorrow morning, thank you.*
  - *Boston, tomorrow*
- *"I wanna go to Boston, tomorrow"*
  - Destination: BOS
  - Departure: 20081028, AM
  - Airline: unspecified
  - Special: unspecified
- *Convert speech to structure*
  - *Sufficient for further processing/query*

Frame-based dialogue agents

1977:

versation with a client who wants to make a simple return trip to a single city in California.

There is good reason for restricting the domain of discourse for a computer system which is t engage in an English dialog. Specializing the subject matter that the system can talk about permi it to achieve some measure of realism without encompassing all the possibilities of human knowledg or of the English language. It also provides the user with specific motivation for participating in th conversation, thus narrowing the range of expectations that GUS must have about the user's pur poses. A system restricted in this way will be more able to guide the conversation within the boundarie of its competence.

Artificial Intelligence Journal, 1977

- ~~Still the industrial state of the art ?~~
- Alexa roughly follows GUS architecture

GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go ?

The state of the art in 1977 !!!!

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

Slot types can be complex, hierarchical



- The type *DATE*

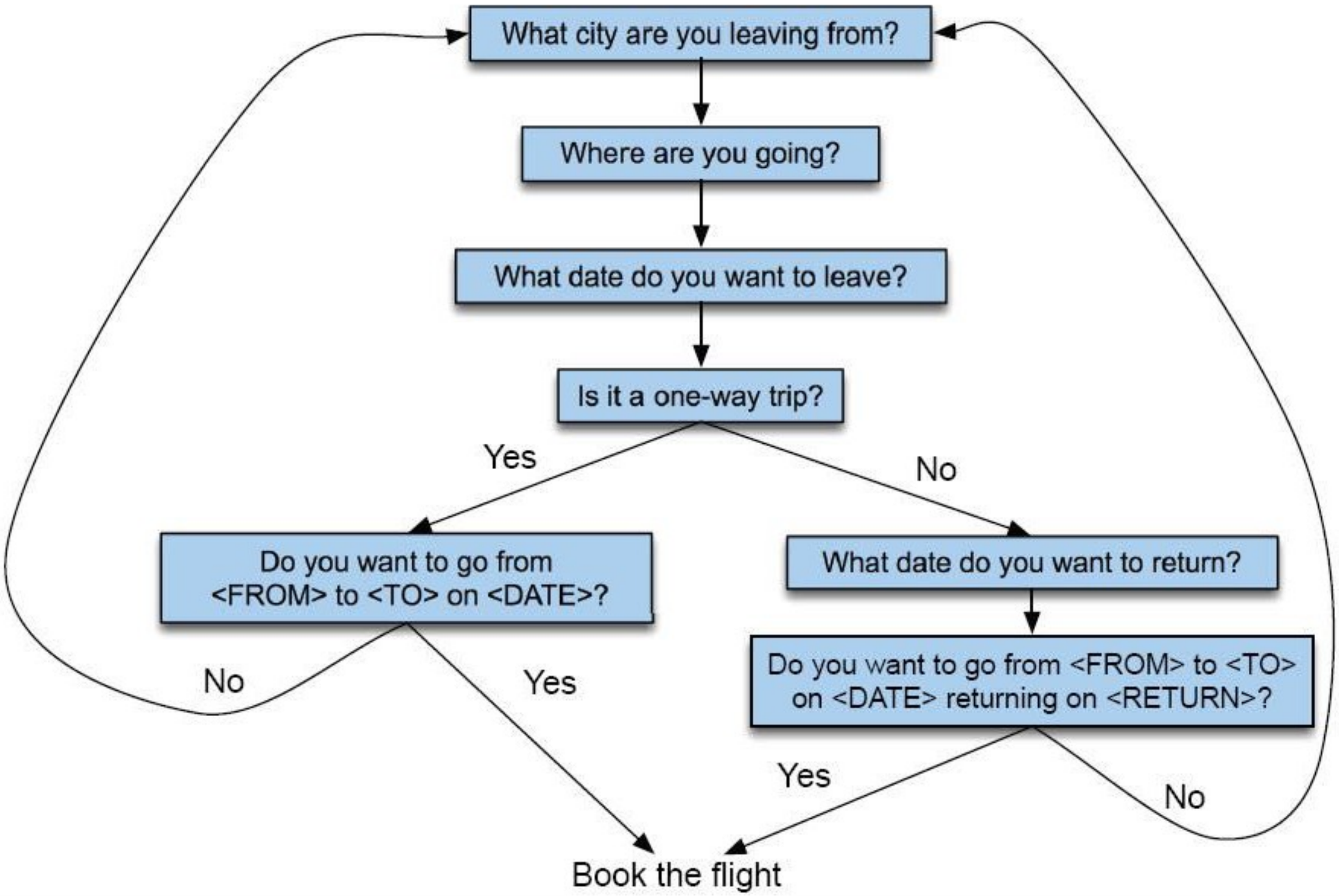
```
DATE
   MONTH NAME
   DAY (BOUNDED-INTEGER 1 31)
   YEAR INTEGER
   WEEKDAY (MEMBER (SUNDAY MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY)]
```

Simple action policy for frame-based dialog



Consider a trivial airline travel system:

- 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


What city are you leaving from?
Where are you going?
What date do you want to leave?
Is it 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
Yes
Yes
No
Book the 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

Frames and mixed initiative



- System asks questions of user, filling any slots that user specifies
  - When frame is filled, do database query
- If user answers 3 questions at once, system can fill 3 slots and not ask these questions again!
- Frame structure guides dialog

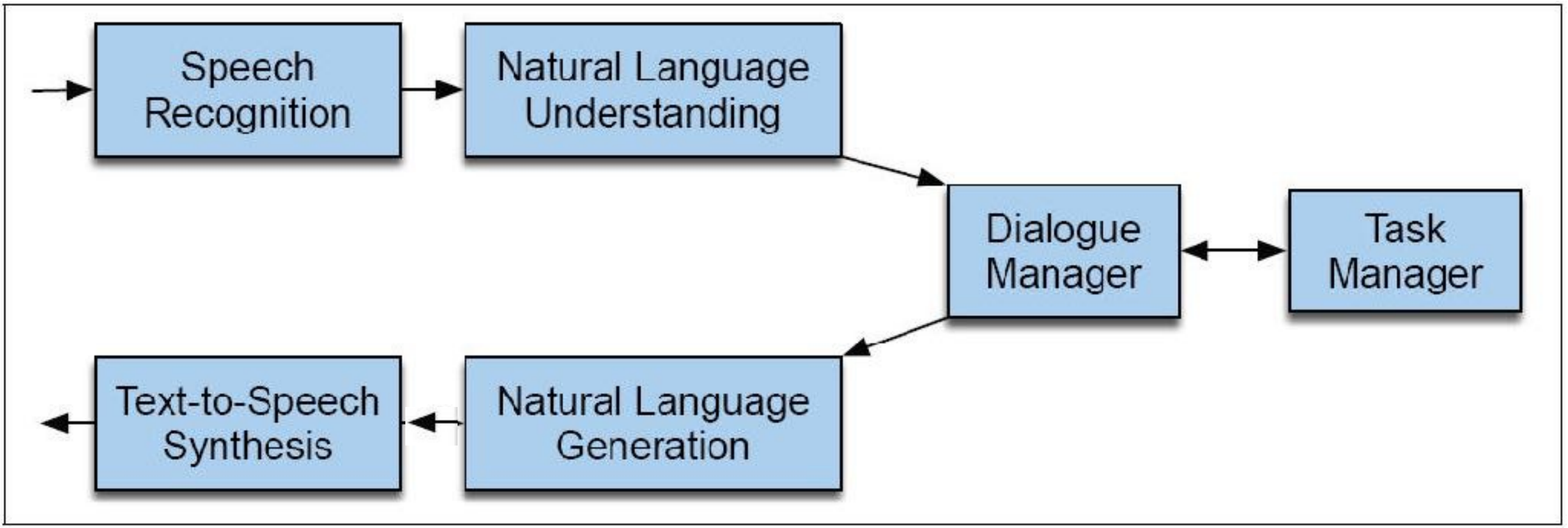
Mixed Initiative



- Conversational initiative can shift between system and user
- Simplest kind of mixed initiative: use the structure of the **frame** to guide dialogue

Slot	Question
ORIGIN	What city are you leaving from?
DESTWhere	are you going?
DEPT DATE	What day would you like to leave?
DEPT TIME	What time would you like to leave?
AIRLINE	What is your preferred airline?

NLU and NLG with frame-based systems



Natural Language Understanding for filling dialog slots

1. Domain classification

   Asking weather? Booking a flight?
   Programming alarm clock?
2. Intent Determination

   Find a Movie, Show Flight, Remove Calendar Appt
3. 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

Wake me tomorrow at six.

```
DOMAIN:   ALARM-CLOCK
INTENT:   SET-ALARM
TIME:     2017-07-01 0600-0800
```

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

Machine learning for slot-filling

I want to fly to San Francisco on Monday afternoon please

Use 1-of-N classifier for Domain/Intent. Use CRF/ sequence model to tag words/phrases with slot names

- Input:
  features like word N-grams
- Output:

  Domain: AIRLINE
  Intent: SHOWFLIGHT
  Destination-City: "San Francisco"
  Depart-Date: "Monday"

More sophisticated algorithm for slot filling: IOB Tagging



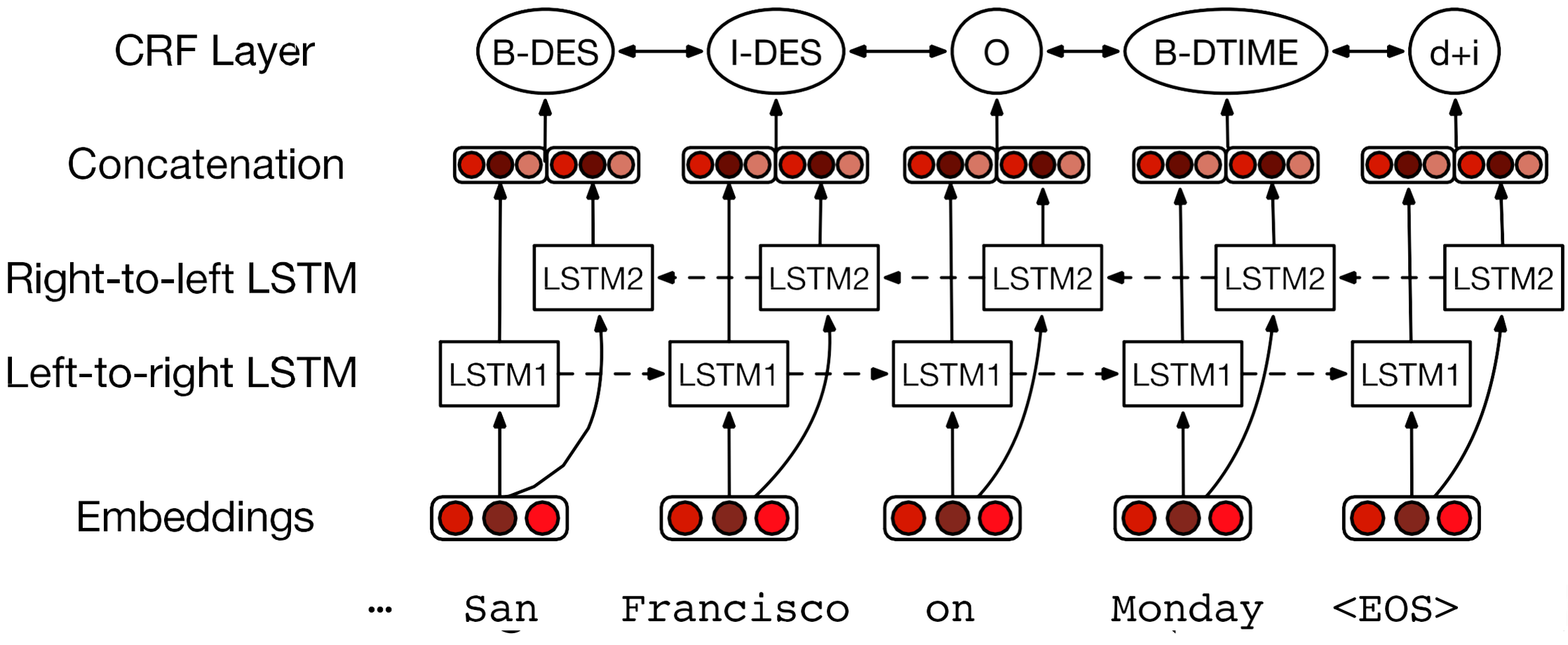
- IOB Tagging
  - tag for the beginning (B) and inside (I) of each slot label,
  - plus one for tokens outside (O) any slot label.
  - $2n + 1$ tags, where n is the number of slots.

B-DESTINASTION
I-DESTINATION
B-DEPART_TIME
I-DEPART_TIME
O

```
O O    O  O   O  B-DES I-DES     O  B-DEPTIME I-DEPTIME  O
I want to fly to San   Francisco on Monday    afternoon  please
```

More sophisticated algorithm for slot filling: IOB Tagging

- IOB Tagging is done by a sequence model
- Typical:



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
- **In practice**: Template-based w/most words prespecified

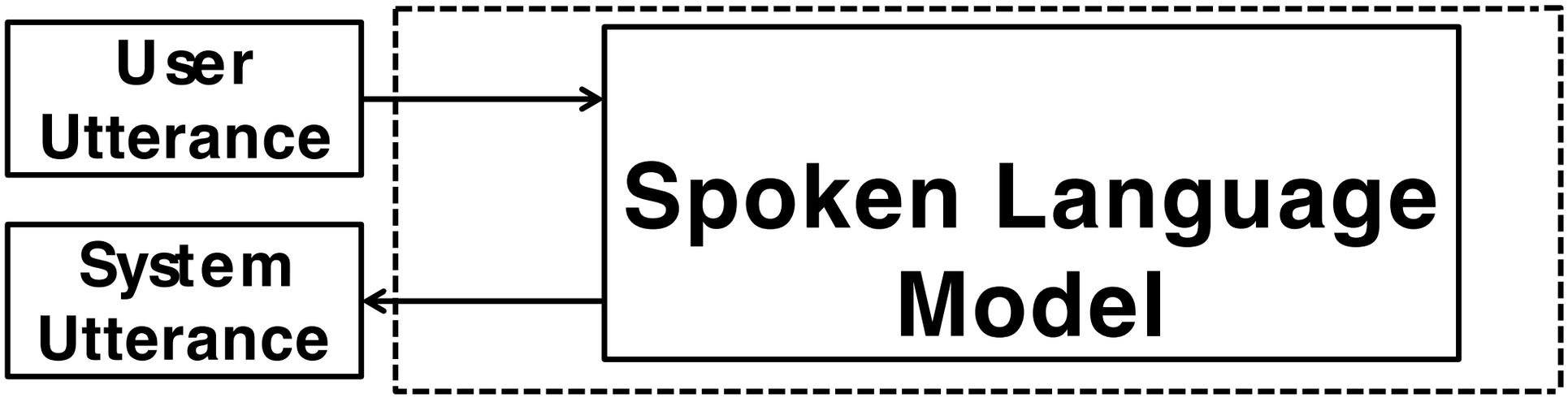
What time do you want to leave CITY-ORIG?
Will you return to CITY-ORIG from CITY-DEST?

More sophisticated NLG



- Dialogue manager builds representation of meaning of utterance to be expressed
- Passes this to a “generator”
- Increasingly generators are deep learning systems
- Mixing chatbot-like NLG constrained to convey dialog representation can improver user satisfaction

End-to-End Frameworks



Duplex Definitions

- Spoken Dialogue Model: AI models which can have a conversation with the user
- Duplex Modes:
  - Half-Duplex: The model waits for one channel to finish talking, before responding
  - Full-Duplex: The model is allowed to simultaneously listen and speak, and can freely interrupt one channel, naturally.
- The benefits of Full-Duplex:
  - Better models for human conversations
  - Better latency and naturalness

Duplex Definitions

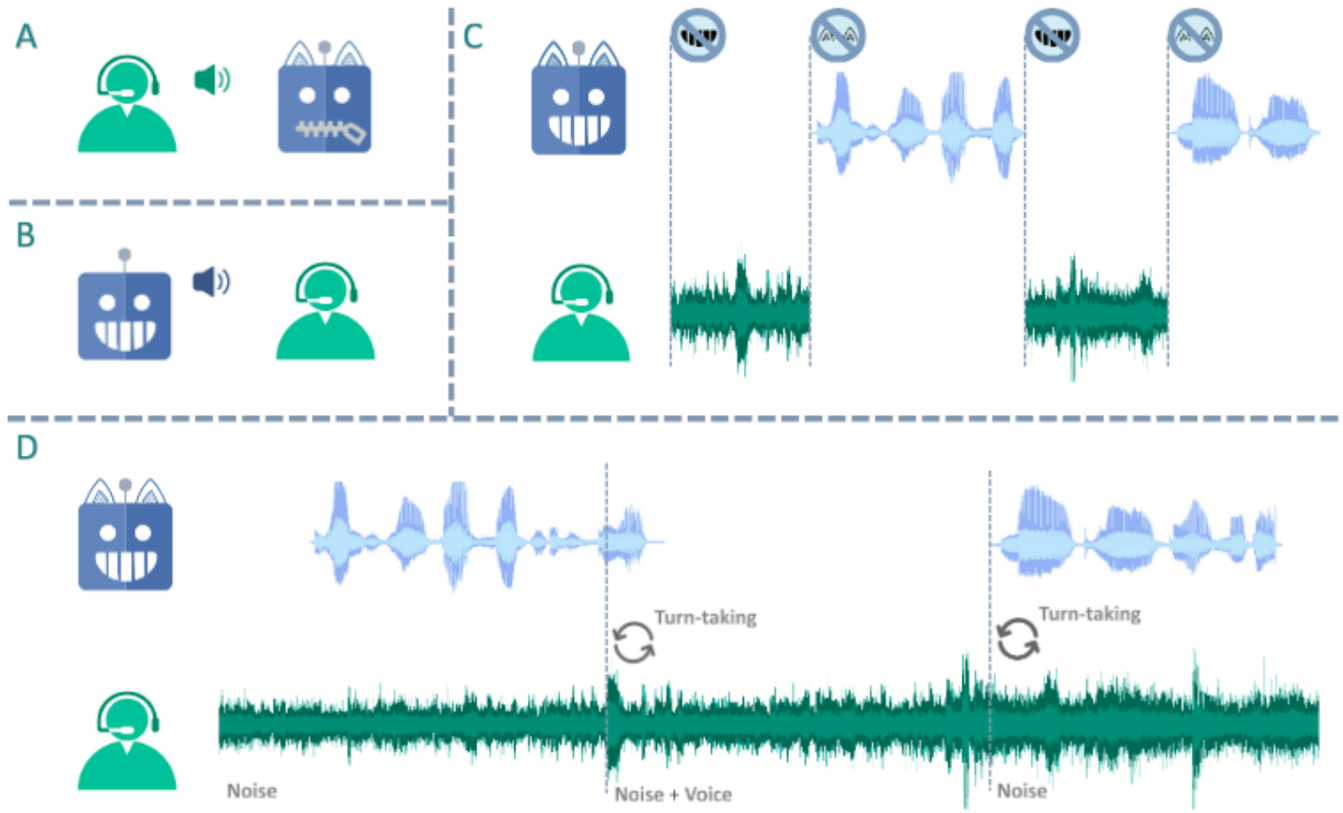


Figure 1: Illustration of simplex, half duplex, and full duplex speech language models. (A): Simplex speech language model with listening ability. (B): Simplex speech language model with speaking ability. (C): Half duplex speech language model with both listening and speaking abilities. (D): Full duplex speech language model can listen while speaking.

Integrating Spoken Dialog into Language Models



Very active Research area recently

- SpeechGPT (May 2023)
- Audio PaLM (June 2023)
- SALMONN (October 2023)
- SpeechVerse (May 2024)
- GPT-4o (May 2024)
- Qwen2-Audio (July 2024)
- Qwen3-Omni (Sep 2025)
- Moshi/ Sesame/ Dessa (Current)

Enabled by discrete representations of speech!

dGSLM (first attempt at Dialog)



Enabled by discrete representations of speech!

Generative Spoken Dialogue Language Modeling

Tu Anh Nguyen[1,3], Eugene Kharitonov[1], Jade Copet[1], Yossi Adi[1], Wei-Ning Hsu[1], Ali Elkahky[1], Paden Tomasello[1], Robin Algayres[1], Benoît Sagot[3], Abdelrahman Mohamed[1], Emmanuel Dupoux[1,2]

Meta AI Research[1], EHESS, Paris[2], Inria, Paris[3]

{ntuanh, abdo, dpx}@fb.com

dGSLM

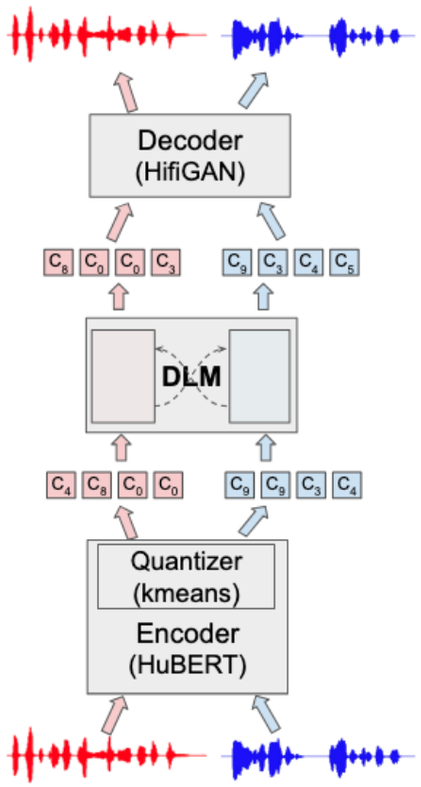


Figure 1: General Schema for dGSLM: A discrete encoder (HuBERT+kmeans) turns each channel of a dialogue into a string of discrete units $(c_1, ..c_N)$. A Dialogue Language Model (DLM) is trained to autoregressively produce units that are turned into waveforms using a decoder (HifiGAN).

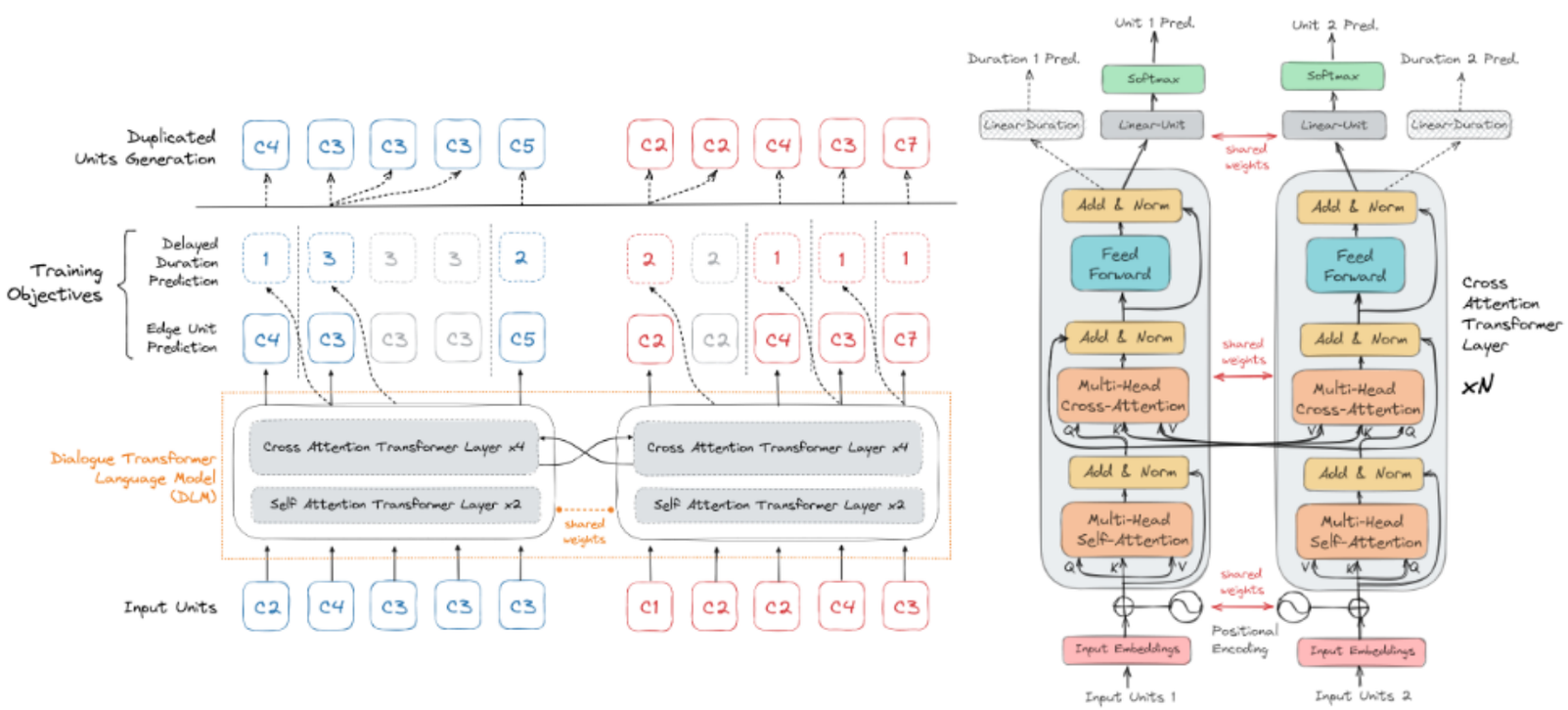


Figure 2: **Illustration of the Dialogue Transformer Language Model (DLM).** Left: DLM Training Objectives. During training, the loss is applied only to edge units and their durations. During generation, the model duplicates the units with the corresponding predicted durations. Right: The Cross-Attention Transformer Layer Architecture.

dGSLM

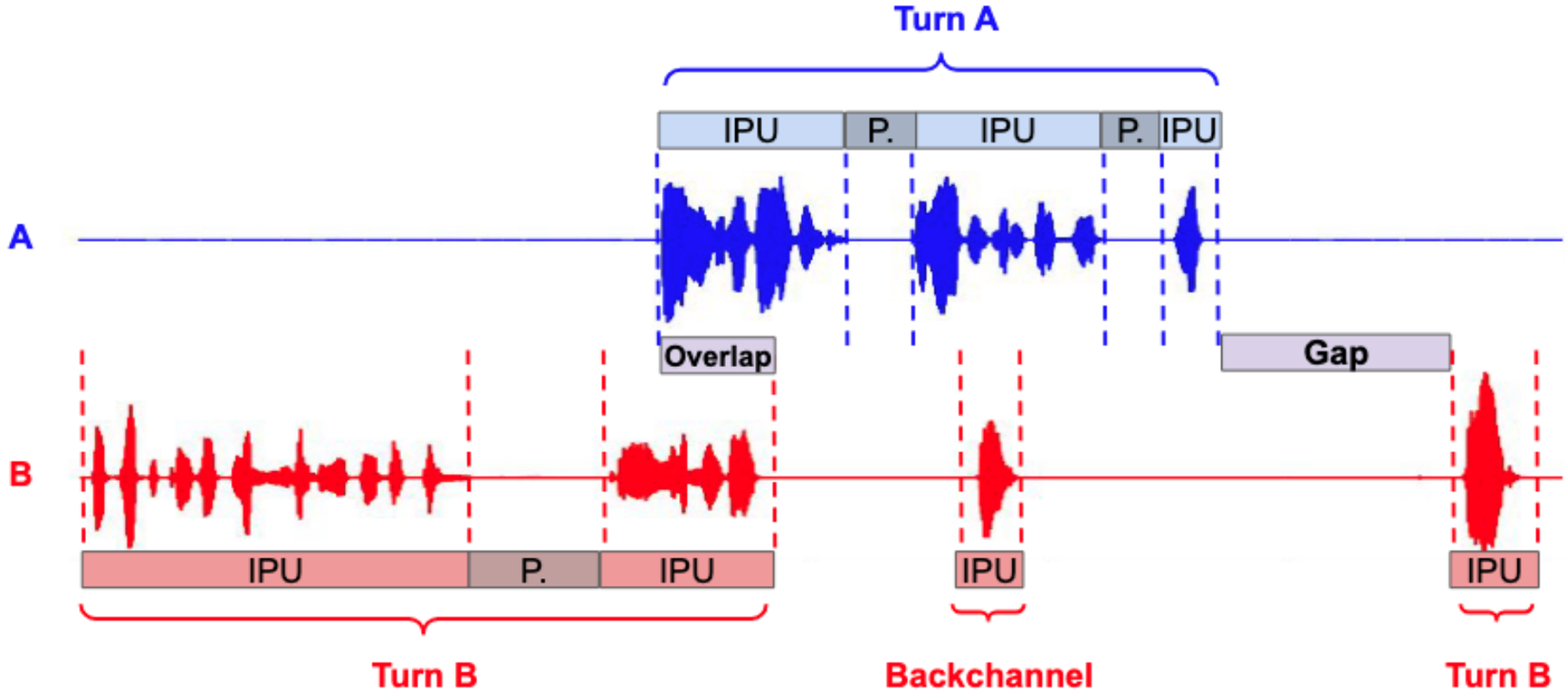


Figure 3: **Illustration of turn-taking events:** IPU (Interpausal Unit), Turn (for speaker A and Speaker B, resp), P. (within-speaker Pause), Gap, Overlap and Backchannel.

dGSLM



- Dialogue Transformer Language Model:
  - 2 tower transformer with Cross Attention between channels
  - Trained with Edge Unit Prediction and Delayed Duration Prediction
  - Essentially predicted the next unit and the duration of the next unit
- Training Data:
  - Only trained on the Fisher Dataset
  - 16,000 dual channel English telephone conversations
  - Averages 10 mins per conversation
  - Over 2000 hours of transcribed speech
  - Trained on 32 V100 32Gb GPUs

Samples



- https://speechbot.github.io/dgslm/

Moshi

Moshi: a speech-text foundation model for real-time dialogue

Alexandre Défossez* ALEX@KYUTAI.ORG
Laurent Mazaré*
Manu Orsini
Amélie Royer
Patrick Pérez
Hervé Jégou
Edouard Grave*
Neil Zeghidour* NEIL@KYUTAI.ORG
Kyutai
**Equal contribution*

Moshi

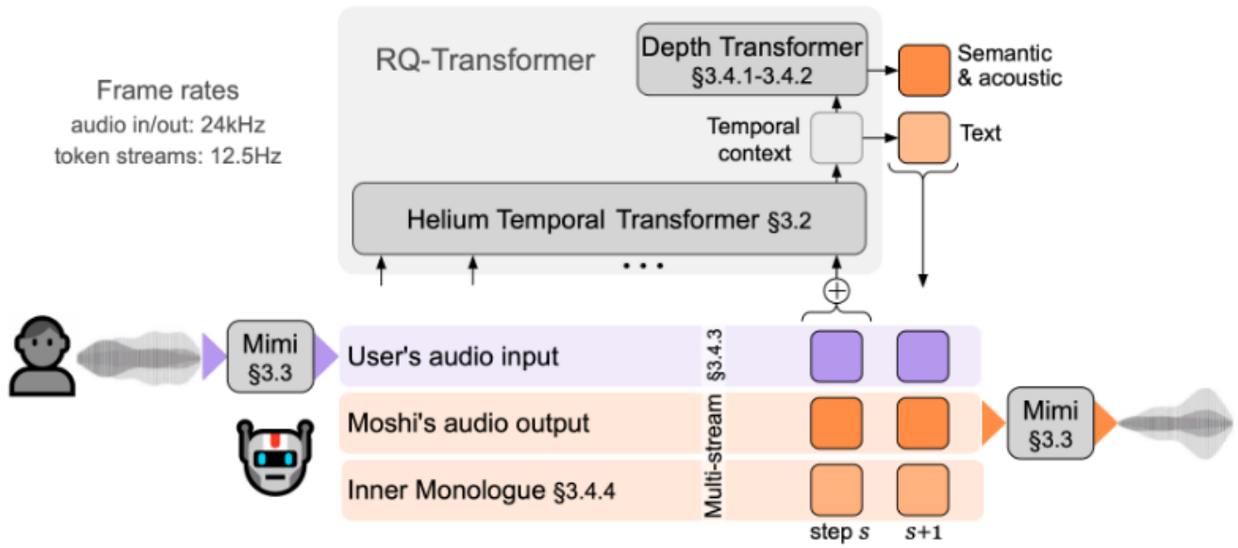


Figure 1: **Overview of Moshi**. Moshi is a speech-text foundation model which enables real-time spoken dialogue. The main components of Moshi's architecture are: a bespoke text language model backbone (Helium, see Section 3.2); a neural audio codec with residual vector quantization and with semantic knowledge distilled from a self-supervised speech model (Mimi, Section 3.3); the streaming, hierarchical generation of semantic and acoustic tokens for both the user and Moshi, along with time-aligned text tokens for Moshi when using Inner Monologue (Section 3.4).

Helium Details



- 7B-parameter autoregressive text LLM, built from scratch
  - Architecture: Standard Transformer (Vaswani et al) with RoPE, RMSNorm, and Gated Linear Units
- Tokenizer based on SentencePiece tokenizer

Moshi

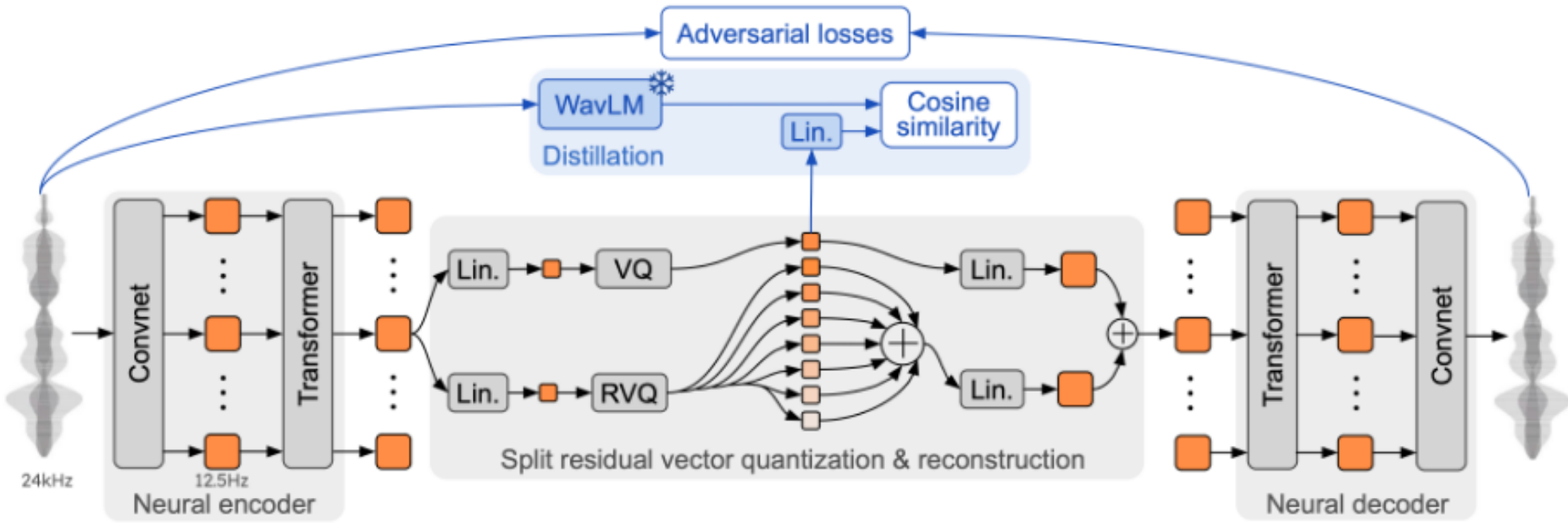


Figure 2: **Architecture and training of Mimi, our neural audio codec, with its split residual vector quantization**. During training (blue part, top), we distill non-causal embeddings from WavLM (Chen et al., 2022) into a single vector quantizer which produces semantic tokens, and is combined with separate acoustic tokens for reconstruction.

Mimi Details



- Causal, streaming audio codec
  - Architecture: “Split RVQ” which consists of 1-level plain VQ (semantics) and 7-level RVQ (acoustic) which are ran in parallel
  - Specs: 8 quantizers with codebook size of 2048. It produces 8 tokens per frame at 12.5 Hz with 1.1kbps
- Similar to SpeechTokenizer, distilled non-causal SSL features from WavLM
- Trained solely with adversarial training

Moshi

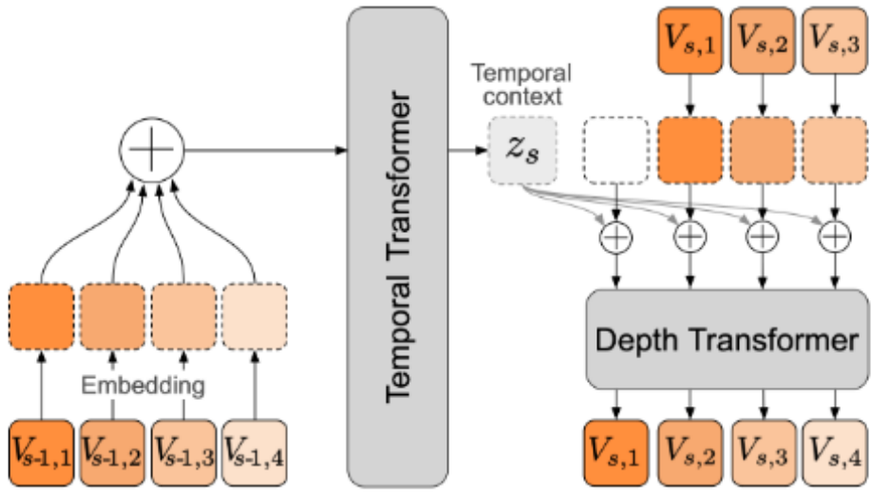


Figure 3: **Architecture of the RQ-Transformer.** The RQ-Transformer breaks down a flattened sequence of length $K \cdot S$ into S timesteps for a large Temporal Transformer which produces a context embedding used to condition a smaller Depth Transformer over K steps. This allows scaling to longer sequences by increasing S—or to a higher depth by increasing K— than modeling the flattened sequence with a single model. In this figure, we use $K = 4$ for the sake of illustration.

Modeling Features

- RQ-Transformer
- Acoustic Delay
- Multi-Stream Modeling
- Inner Test Monologue

Joint sequence modeling for Moshi. Putting together the multi-stream and inner monologue, we have the final set V of sequences to model defined as

$$\begin{cases} V_{s,1} & = W_s & & & \text{aligned text tokens.} \\ V_{s,2} & = A_{s,1} & & & \text{semantic tokens of Moshi.} \\ V_{s,1+q} & = A_{s-\tau,q} & \text{if} & s \geq \tau+1, 1 < q \leq Q & \text{delayed acoustic tok. of Moshi.} \\ V_{s,1+Q+1} & = A'_{s,1} & & & \text{semantic tokens of } \textit{other}. \\ V_{s,1+Q+q} & = A'_{s-\tau,q} & \text{if} & s \geq \tau+1, 1 < q \leq Q & \text{delayed acoustic tok. of } \textit{other}, \end{cases} \quad (6)$$

amounting to a total number of $K = 2Q + 1$ streams, with $Q = 8$ in the experiments. A summary is provided in Figure 4.

Moshi

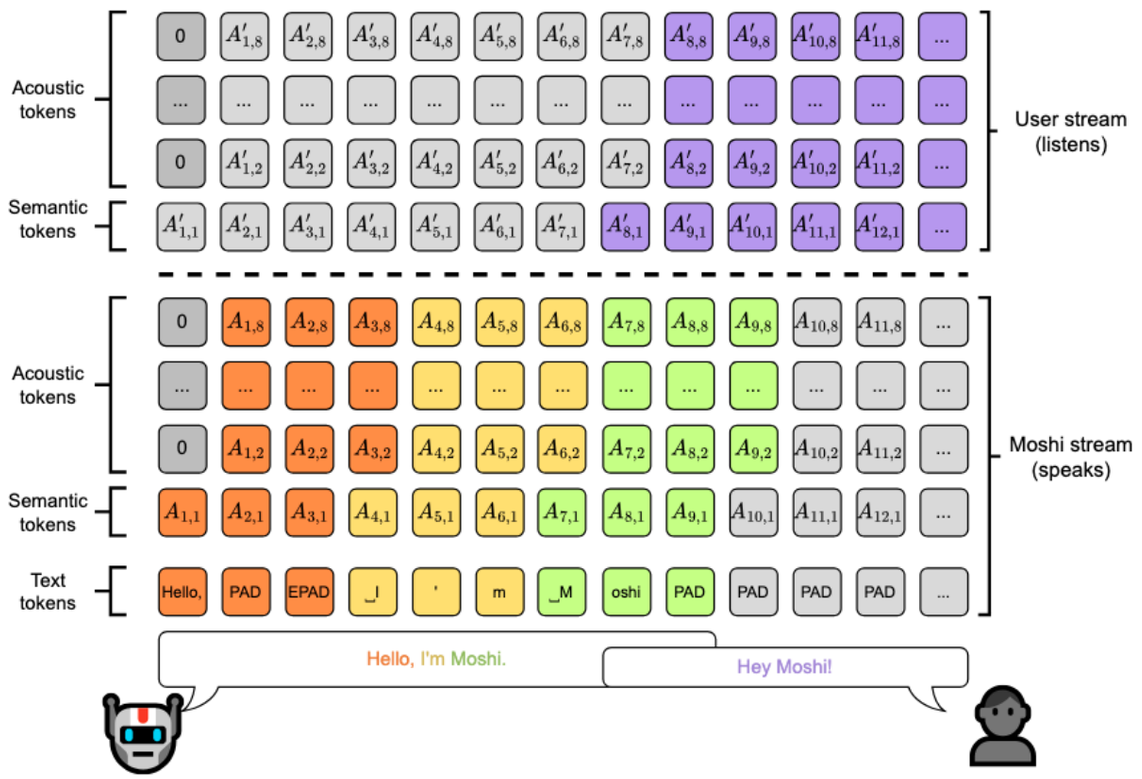


Figure 4: **Representation of the joint sequence modeled by Moshi**. Each column represents the tokens for a given step in the joint sequence ($V_{s,k}$) described in Equation 6 with an acoustic delay $\tau = 1$, e.g. the input of the Temporal Transformer for this step. Tokens are predicted from bottom to top in the Depth Transformer. At inference time, tokens under the dashed line (corresponding to Moshi) are sampled, while those above are fed from the user. This design allows for our model to handle overlapping speech turns.

Moshi Details



- Helium pre-training: trained on 2.1 trillion text tokens
- Moshi 4-Stage Training:
  - Pre-training: Temporal Transformer init. with Helium. Trained on 7M hours of single-stream unsupervised audio + Helium pre-training dataset
  - Post-training: Trained on the same 7M hours of audio, but used PyAnnote diarization to simulate a 2-stream conversation.
  - Finetuning: Finetuned on the 2,000-hour Fisher dataset, which has true separate channels per speaker, to learn real overlapping speech.
  - Instruct Finetuning: Finetuned on 20k+ hours of synthetic data generated by:
    - Using the Helium LLM to generate realistic conversation transcripts
    - Synthesizing these transcripts with a separate TTS model (which itself was trained on 170h of private data)



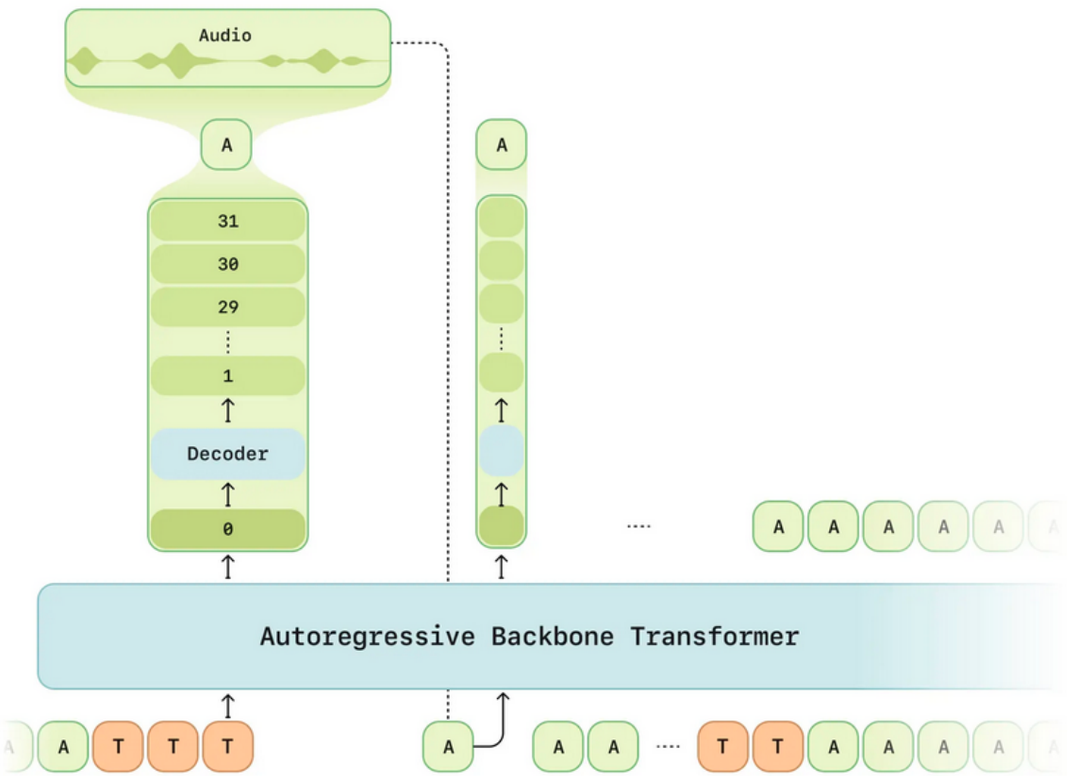
Crossing the uncanny valley of conversational voice

February 27, 2025

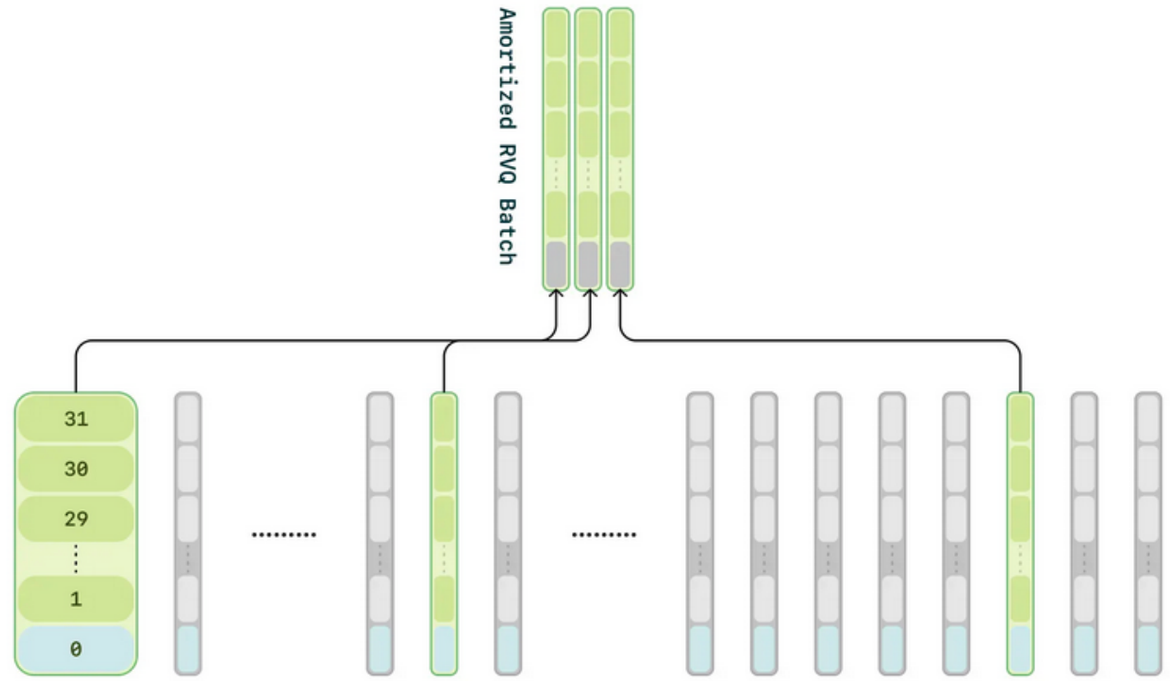
Brendan Iribe, Ankit Kumar, and the Sesame team

Brendan Iribe, Ankit Kumar, Ryan Brown,
Angela Gayles, Hans Hartmann, Nate Mitchell

Sesame



CSM model inference process. Text (T) and audio (A) tokens are interleaved and fed sequentially into the Backbone, which predicts the zeroth level of the codebook. The Decoder then samples levels 1 through N – 1 conditioned on the predicted zeroth level. The reconstructed audio token (A) is then autoregressively fed back into the Backbone for the next step, continuing until the audio EOT symbol is emitted. This process begins again on the next inference request, with the interim audio (such as a user utterance) being represented by interleaved audio and text transcription tokens.



Amortized training process. The backbone transformer models the zeroth level across all frames (highlighted in blue), while the decoder predicts the remaining N – 31 levels, but only for a random 1/16th of the frames (highlighted in green). The top section highlights the specific frames modeled by the decoder for which it receives loss.

Technical Report Summary



- Does not contain all of the info about their system, just a subset
- Trained a RQ-Transformer on Mimi (Moshi's codec, 12.5 Hz)
- Trained on 1 million hours of English data
- Trained a Tiny 1B model up to a 8B model
- Open sourced 1B model is a pure Mimi based Audio Language model finetuned on Llama-3.2-1B
- https://huggingface.co/spaces/sesame/csm-1b
- Data format is interesting, it is basically just an interwoven text-speech transformer:
  - [0]Hello how are you doing[mimi tokens for this speech]
  - [1]I am doing fine[model generates mini tokens for this speech]

Dialog Research Challenges



- History or context tracking
- Not enough data
- Big variance across different dialog domains
- Difficult to evaluate moving targets (human)!