Challenges in Dialog & Dialog Systems

Ling575
Discourse and Dialog
May 11, 2011

Roadmap

- Issues in Dialog & Dialog Systems
  - Linguistics of Conversation
    - Grounding
    - Implicature
  - Dialog Systems
    - Architecture
    - Components
    - Evaluation

Collaborative Communication

- Speaker tries to establish and add to
  - “common ground” – “mutual belief”

  - Presumed a joint, collaborative activity
    - Make sure “mutually believe” the same thing

  - Hearer must ‘ground’ speaker’s utterances
    - Indicate heard and understood

Closure

- Principle of closure:
  - Agents performing an action require evidence of successful performance

Closure

- Principle of closure:
  - Agents performing an action require evidence of successful performance
    - Also important to indicate failure or understanding

Closure

- Principle of closure:
  - Agents performing an action require evidence of successful performance
    - Also important to indicate failure or understanding
  - Non-speech closure:

Closure

- Principle of closure:
  - Agents performing an action require evidence of successful performance
    - Also important to indicate failure or understanding
  - Non-speech closure:
    - Push elevator button -> Light turns on
  - Two step process:
    - Presentation (speaker)
    - Acceptance (listener)

Degrees of Grounding

- Weakest to strongest
- Continued attention:
  - Silence implies consent
- Next relevant contribution
- Acknowledgment:
  - Minimal response, continuer: *yeah, uh-huh, okay; great*
- Demonstrate:
  - Indicate understanding by reformulation, completion
- Display:
  - Repeat all or part

Grounding Examples

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 in HCI

- Key factor in HCI:
  - Users confused if system fails to ground, confirm
    - (Stifelman et al., 1993), (Yankelovich et al, 1995)

Grounding in HCI

- Key factor in HCI:
  - Users confused if system fails to ground, confirm
    - (Stifelman et al., 1993), (Yankelovich et al, 1995)

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

Grounding in HCI

- Key factor in HCI:
  - Users confused if system fails to ground, confirm
    - (Stifelman et al., 1993), (Yankelovich et al, 1995)

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

    - S: Did you want to review some more of your profile?
    - U: No.
    - S: What's next?

Computational Models

- (Traum et al) revised for computation
  - Involves both speaker and hearer
    - Initiate, Continue, Acknowledge, Repair, Request Repair, etc

Computational Models

- (Traum et al) revised for computation
  - Involves both speaker and hearer
    - Initiate, Continue, Acknowledge, Repair, Request Repair, etc
  - Common phenomena
    - "Back-Channel" – "uh-huh", "okay", etc
      - Allows hearer to signal continued attention, ack
        - WITHOUT taking the turn

Computational Models

- (Traum et al) revised for computation
  - Involves both speaker and hearer
    - Initiate, Continue, Acknowledge, Repair, Request Repair, etc
  - Common phenomena
    - “Back-Channel” – “uh-huh”, “okay”, etc
      - Allows hearer to signal continued attention, ack
        - WITHOUT taking the turn
    - Requests for repair – common in human-human
      - Even more common in human-computer dialogue

Conversational Structure

- Structure beyond adjacency pairs

Conversational Structure

- Structure beyond adjacency pairs
- Openings and closings (Clark, 1994; Schegloff, 1968)
    - Stage 1: Enter with summons-response
    - Stage 2: Identification
    - Stage 3: Establish willingness
    - Stage 4: Initiate first topic (caller)

Stage	Speaker & Utterance
1	A_1: (rings B's telephone)
1,2	B_1: Benjamin Holloway
2	A_1: this is Professor Dwight's secretary, from Polymania College
2,3	B_1: ooh yes –
4	A_1: uh:m . about the: lexicology *seminar*
4	B_1: *yes*

Conversational Implicature

- Meaning more than just literal contribution

  - *A: And, what day in May did you want to travel?*
  - *C: OK uh I need to be there for a meeting the 12-15th*
    - Appropriate?

Conversational Implicature

- Meaning more than just literal contribution

  - *A: And, what day in May did you want to travel?*
  - *C: OK uh I need to be there for a meeting the 12-15th*
    - Appropriate? Yes
    - Why?

Conversational Implicature

- Meaning more than just literal contribution

  - *A: And, what day in May did you want to travel?*
  - *C: OK uh I need to be there for a meeting the 12-15th*
    - Appropriate? Yes
    - Why?
  - A: ... *There's three non-stops today.*
    - Are there 4?

Conversational Implicature

- Meaning more than just literal contribution

  - *A: And, what day in May did you want to travel?*
  - *C: OK uh I need to be there for a meeting the 12-15th*
    - Appropriate? Yes
    - Why?
  - A: … *There's three non-stops today.*
    - Are there 4? No
    - How can we tell?
  - Inference guides

Grice's Maxims

- Cooperative principle:
  - Tacit agreement b/t conversants to cooperate
- Grice's Maxims
  - Quantity: Be as informative as required
  - Quality: Be truthful
    - Don't lie, or say things without evidence
  - Relevance: Be relevant
  - Manner: "Be perspicuous"
    - Don't be obscure, ambiguous, prolix, or disorderly

Relevance

- A: **Is Regina here?**
- B: **Her car is outside**.

Relevance

- A: **Is Regina here?**
- B: **Her car is outside**.
- Implication: yes
  - Hearer thinks: **why would he mention the car? It must be relevant. How could it be relevant? It could since if her car is here she is probably here.**
- Client: **I need to be there for a meeting that's from the 12th to the 15th**

Relevance

- A: **Is Regina here?**
- B: **Her car is outside**.
- Implication: yes
  - Hearer thinks: **why would he mention the car? It must be relevant. How could it be relevant? It could since if her car is here she is probably here.**
- Client: **I need to be there for a meeting that's from the 12th to the 15th**
  - Hearer thinks: **Speaker is following maxims, would only have mentioned meeting if it was relevant. How could meeting be relevant? If client meant me to understand that he had to depart in time for the mtg.**

Quantity

- A:How much money do you have on you?
- B: I have 5 dollars

Quantity

- A:How much money do you have on you?
- B: I have 5 dollars
  - Implication: not 6 dollars
- Similarly, 3 non stops can't mean 7 non-stops (hearer thinks:
  - if speaker meant 7 non-stops she would have said 7 non-stops
- A: Did you do the reading for today's class?
- B: I intended to
  - Implication:

Quantity

- A:How much money do you have on you?
- B: I have 5 dollars
  - Implication: not 6 dollars
- Similarly, 3 non stops can't mean 7 non-stops (hearer thinks:
  - if speaker meant 7 non-stops she would have said 7 non-stops
- A: Did you do the reading for today's class?
- B: I intended to
  - Implication: No
  - B's answer would be true if B intended to do the reading AND did the reading, but would then violate maxim

Grice's Maxims

- "Flouting" maxims:
  - Consciously violate for effect
    - Humor, emphasis
      - There must be a *million* flights!

Implicature in SDS

- System should produce output consistent w/maxims
- System should/can (mostly) assume:
  - User utterances related to current task
    - Relevance
  - User is accurate/truthful
    - Quality
  - User is sufficiently precise/specific
    - Quantity
  - Etc

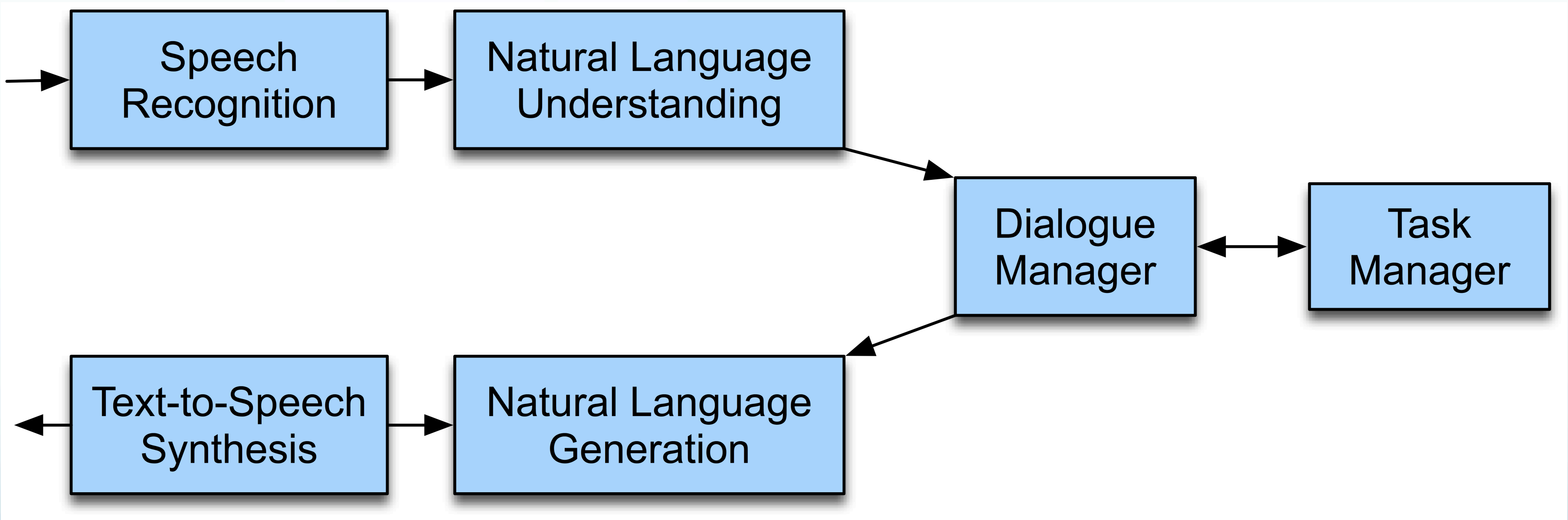
From Human to Computer

- Conversational agents
  - Systems that (try to) participate in dialogues
  - Examples: Directory assistance, travel info, weather, restaurant and navigation info
- Issues:

From Human to Computer

- Conversational agents
  - Systems that (try to) participate in dialogues
  - Examples: Directory assistance, travel info, weather, restaurant and navigation info
- Issues:
  - Limited understanding: ASR errors, interpretation
  - Computational costs:
    - broader coverage -> slower, less accurate

Dialogue System Architecture



Speech Recognition

- (aka ASR)
- Input: acoustic waveform
  - Telephone, microphone, and smartphone
- Output: recognized word string
- Requirements:
  - Acoustic models: map acoustics to phone [ae] [k]
  - Pronunciation dictionary: words to phones: cat: [k][ae][t]
  - Grammar: legal word sequences
  - Search procedure: best word sequence given audio

Speech Recognition

- (aka ASR)
- Input: acoustic waveform
  - Telephone, microphone, and smartphone
- Output: recognized word string

Speech Recognition

- (aka ASR)
- Input: acoustic waveform
  - Telephone, microphone, and smartphone
- Output: recognized word string
- Requirements:
  - Acoustic models: map acoustics to phone [ae] [k]
  - Pronunciation dictionary: words to phones: cat: [k][ae][t]
  - Grammar: legal word sequences
  - Search procedure: best word sequence given audio

Specialization & Restriction

- What does the ASR component need to recognize?
  - Everything

Specialization & Restriction

- What does the ASR component need to recognize?
  - Everything? Not really
  - Only:
    - In-domain utterances
    - Utterances interpretable by understanding component
- How much should the recognizer recognize?
  - Any in-domain utterance any time?

Specialization & Restriction

- What does the ASR component need to recognize?
  - Everything? Not really
  - Only:
    - In-domain utterances
    - Utterances interpretable by understanding component
- How much should the recognizer recognize?
  - Any in-domain utterance any time?
  - Only appropriate responses to last system utterance
    - Restrictive grammar
- Why? Why not?

Recognition in SDS

- Create domain specific vocabulary, grammar
  - Typically hand-crafted in most commercial systems
  - Based on human-human interactions
  - Grammars: finite-state, context-free, language model

Recognition in SDS

- Create domain specific vocabulary, grammar
  - Typically hand-crafted in most commercial systems
  - Based on human-human interactions
  - Grammars: finite-state, context-free, language model

- Activate only portion of grammar based on dialog state
  - E.g. Where are you leaving from?
  - {I want to (leave|depart) from} CITYNAME {STATENAME}
  - ‘Yes/No’ grammar for confirmations

Natural Language Understanding

- Most systems use frame-slot semantics
  Show me morning flights from Boston to SFO on Tuesday

  - SHOW:
  - FLIGHTS:
    - ORIGIN:
      - CITY: Boston
      - DATE:
        - DAY-OF-WEEK: Tuesday
      - TIME:
        - PART-OF-DAY: Morning
    - DEST:
      - CITY: San Francisco

Semantic Grammars

- Alternatives:
  - Full parser with semantic attachments
  - Domain-specific analyzers
- CFG in which the LHS of rules is a semantic category:
  - LIST -> show me | I want | can I see|…
  - DEPARTTIME -> (after|around|before) HOUR| morning | afternoon | evening
  - HOUR -> one|two|three…|twelve (am|pm)
  - FLIGHTS -> (a) flight|flights
  - ORIGIN -> from CITY
  - DESTINATION -> to CITY
  - CITY -> Boston | San Francisco | Denver | Washington

Result

- **SHOW FLIGHT ORIGIN DEST DEP_DATE DEP_TIME**
- *Show me flights from Boston to SFO on Tuesday morning*

Other NLU

- Issues:
  - Ambiguity:
    - Probabilistic CFGs
  - Manual construction:
    - Train HMM analysis model
    - Where's the training data from???

Generation and TTS

- Generation:
  - Identify concepts to express
  - Convert to words
  - Assign appropriate prosody, intonation

Generation and TTS

- Generation:
  - Identify concepts to express
  - Convert to words
  - Assign appropriate prosody, intonation
- TTS:
  - Input words, prosodic markup
  - Synthesize acoustic waveform

Generation

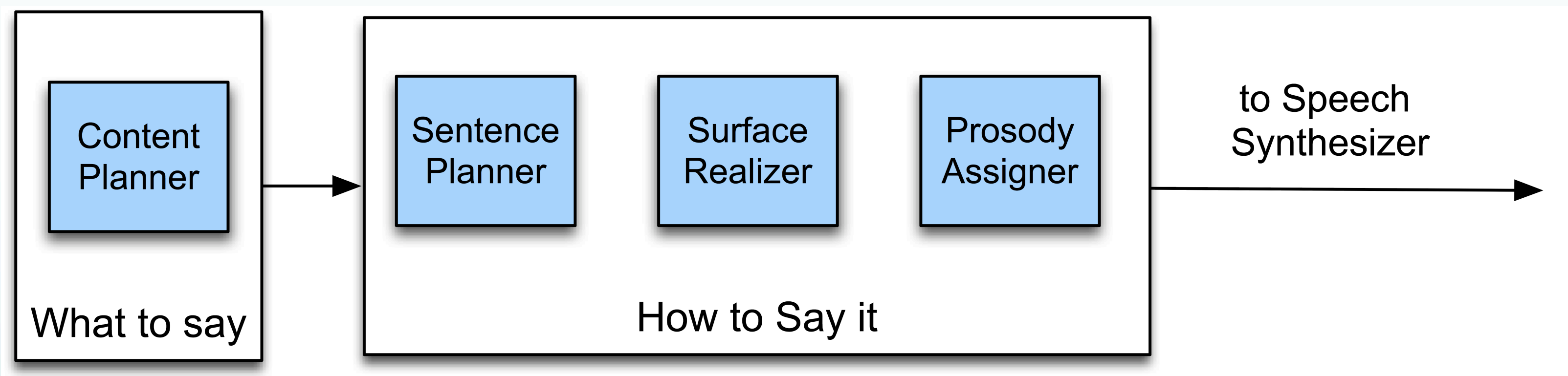
- Content planning:
  - What to say:
    - Question, answer, etc?
    - Often merged with dialog manager

Generation

- Content planning:
  - What to say:
    - Question, answer, etc?
    - Often merged with dialog manager
- Language generation:
  - How to say it
    - Select syntactic structure and words
    - Most common: Template-based generation (prompts)
      - Templates with variable: When do you want to leave CITY?

Full NLG

- Converts representation from dialog manager



SDS Generation Constraints

- Establishing coherence:
  - What tools?

SDS Generation Constraints

- Establishing coherence:
  - What tools? Discourse markers, pronouns
    - ‘Okay’ – signal new topic

    - *Please say the date;*
    - *Please say start time;*
    - *Please say…*

SDS Generation Constraints

- Establishing coherence:
  - What tools? Discourse markers, pronouns
    - ‘Okay’ – signal new topic

    - *Please say the date;*
    - *Please say start time;*
    - *Please say…*
    - vs
    - *First tell me the date;*
    - *Thanks ;; Next I’ll need the start time;*
    - *Lastly, I’ll ….*

Handling SDS Generation

- Key issues:

Handling SDS Generation

- Key issues:
  - Speech is slow!
  - Speech is not persistent!
- Slowness: Minimize prompt length, repetition

Handling SDS Generation

- Key issues:
  - Speech is slow!
  - Speech is not persistent!
- Slowness: Minimize prompt length, repetition
  - Approach: Tapered prompts
    - What's the first company? What's the next company? Next company? Next?
- Slowness/transience:

Handling SDS Generation

- Key issues:
  - Speech is slow!
  - Speech is not persistent!
- Slowness: Minimize prompt length, repetition
  - Approach: Tapered prompts
    - What’s the first company? What’s the next company? Next company? Next?
- Slowness/transience:
  - Handling lists, long material
    - Email
    - Rules if list is long (>3); simplify, give one result

Dialogue Manager

- Holds system together: Governs interaction style
  - Takes input from ASR/NLU
  - Maintains dialog state, history
    - Incremental frame construction
    - Reference, ellipsis resolution
    - Determines what system does next
  - Interfaces with task manager/backend app
  - Formulates basic response, passes to NLG,TTS

Dialog Management Types

- Finite-State Dialog Management
- Frame-based Dialog Management
- Information State Manager
- AI Planning System

Finite-State Dialogue Management

- Simplest type of dialogue management
  - States:
  - Arcs:

Finite-State Dialogue Management

- Simplest type of dialogue management
  - States:
    - Questions system asks user
  - Arcs:

Finite-State Dialogue Management

- Simplest type of dialogue management
  - States:
    - Questions system asks user
  - Arcs:
    - User responses
- System controls interactions:

Finite-State Dialogue Management

- Simplest type of dialogue management
  - States:
    - Questions system asks user
  - Arcs:
    - User responses
- System controls interactions:
  - Interprets all input based on current state
  - Assumes any user input is response to last question

Finite-State Dialogue Management

- Initiative:
  - Control of the interaction
- Who's in control here?

Finite-State Dialogue Management

- Initiative:
  - Control of the interaction
- Who’s in control here?
  - System!
    - “system initiative”/”single initiative”

Finite-State Dialogue Management

- Initiative:
  - Control of the interaction
- Who’s in control here?
  - System!
    - “system initiative”/”single initiative”
  - Natural?

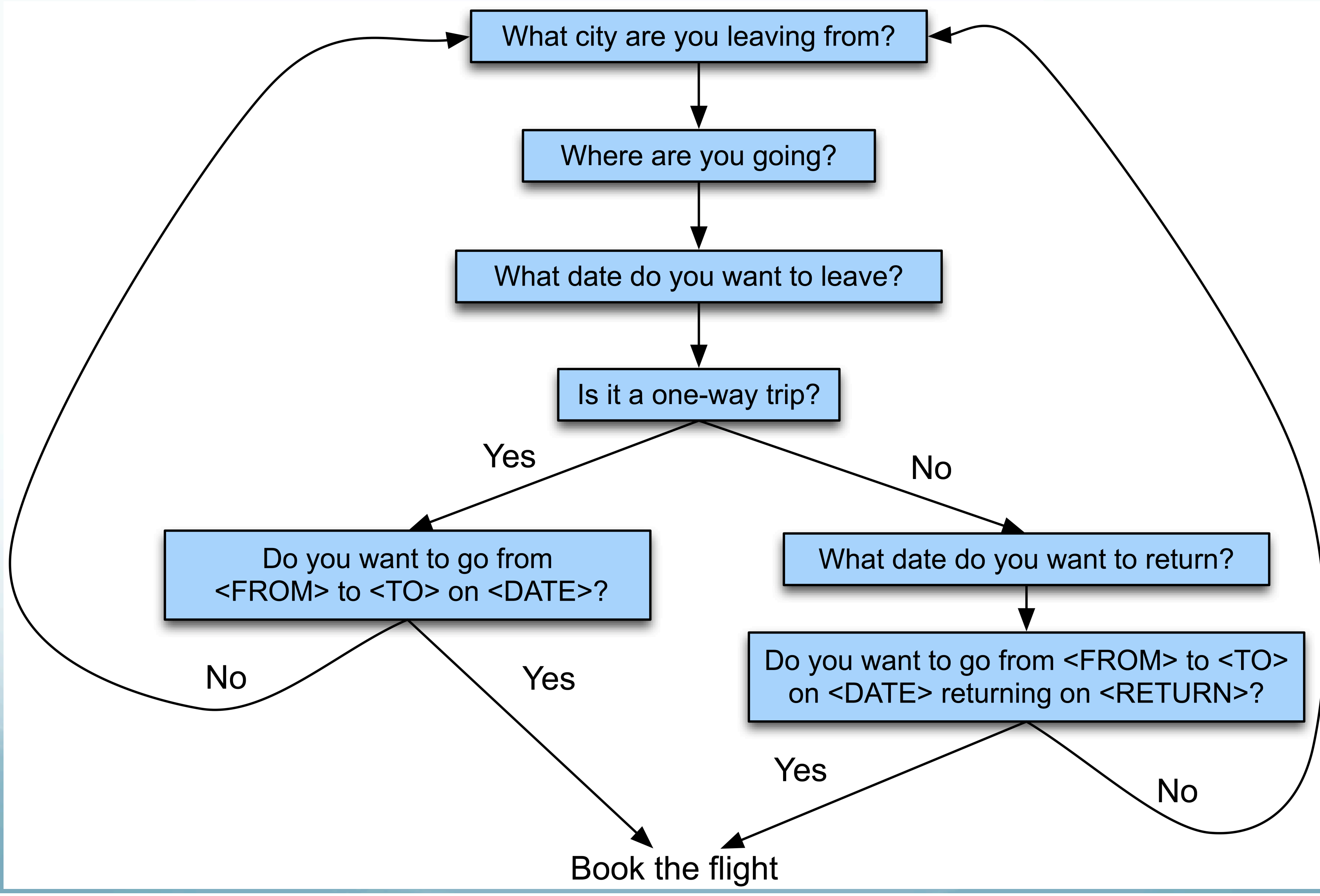
Finite-State Dialogue Management

- Initiative:
  - Control of the interaction
- Who's in control here?
  - System!
    - "system initiative"/"single initiative"
  - Natural? No!
    - Human conversation goes back and forth
- Deploy targeted vocabulary / grammar for state

Finite-State Dialogue Management

- Initiative:
  - Control of the interaction
- Who's in control here?
  - System!
    - "system initiative"/"single initiative"
  - Natural? No!
    - Human conversation goes back and forth
- Deploy targeted vocabulary / grammar for state
  - Add 'universals' – accessible anywhere in dialog
    - 'Help', 'Start over'

Finite-State Management



Pros and Cons

- Advantages

Pros and Cons

- Advantages
  - Straightforward to encode
  - Clear mapping of interaction to model
  - Well-suited to simple information access
  - System initiative
- Disadvantages

Pros and Cons

- Advantages
  - Straightforward to encode
  - Clear mapping of interaction to model
  - Well-suited to simple information access
  - System initiative
- Disadvantages
  - Limited flexibility of interaction
    - Constrained input – single item
    - Fully system controlled
    - Restrictive dialogue structure, order
  - Ill-suited to complex problem-solving

Frame-based Dialogue Management

- Finite-state too limited, stilted, irritating
- More flexible dialogue

Frame-based Dialogue Management

- Essentially form-filling
  - User can include any/all of the pieces of form
  - System must determine which entered, remain

Frame-based Dialogue Management

- Essentially form-filling
  - User can include any/all of the pieces of form
  - System must determine which entered, remain

Slot	Question
ORIGIN CITY	"From what city are you leaving?"
DESTINATION CITY	"Where are you going?"
DEPARTURE TIME	"When would you like to leave?"
ARRIVAL TIME	"When do you want to arrive?"

Frame-based Dialogue Management

- Essentially form-filling
  - User can include any/all of the pieces of form
  - System must determine which entered, remain

Slot	Question
ORIGIN CITY	"From what city are you leaving?"
DESTINATION CITY	"Where are you going?"
DEPARTURE TIME	"When would you like to leave?"
ARRIVAL TIME	"When do you want to arrive?"

- System may have multiple frames
  - E.g. flights vs restrictions vs car vs hotel
  - Rules determine next action, question, information presentation

Frames and Initiative

- Mixed initiative systems:
  - A) User/System can shift control arbitrarily, any time
    - Difficult to achieve

Frames and Initiative

- Mixed initiative systems:
  - A) User/System can shift control arbitrarily, any time
    - Difficult to achieve
  - B) Mix of control based on prompt type
- Prompts:

Frames and Initiative

- Mixed initiative systems:
  - A) User/System can shift control arbitrarily, any time
    - Difficult to achieve
  - B) Mix of control based on prompt type
- Prompts:
  - Open prompt:

Frames and Initiative

- Mixed initiative systems:
  - A) User/System can shift control arbitrarily, any time
    - Difficult to achieve
  - B) Mix of control based on prompt type
- Prompts:
  - Open prompt: ‘How may I help you?’
    - Open-ended, user can respond in any way
  - Directive prompt:

Frames and Initiative

- Mixed initiative systems:
  - A) User/System can shift control arbitrarily, any time
    - Difficult to achieve
  - B) Mix of control based on prompt type
- Prompts:
  - Open prompt: 'How may I help you?'
    - Open-ended, user can respond in any way
  - Directive prompt: 'Say yes to accept call, or no o.w.'
    - Stipulates user response type, form

Initiative, Prompts, Grammar

- Prompt type tied to active grammar
  - System must recognize suitable input
    - Restrictive vs open-ended

Initiative, Prompts, Grammar

- Prompt type tied to active grammar
  - System must recognize suitable input
    - Restrictive vs open-ended
- Shift from restrictive to open
  - Tune to user: Novice vs Expert

Initiative, Prompts, Grammar

- Prompt type tied to active grammar
  - System must recognize suitable input
    - Restrictive vs open-ended
- Shift from restrictive to open
  - Tune to user: Novice vs Expert

	Prompt Type	
Grammar	**Open**	**Directive**
Restrictive	*Doesn't make sense*	System Initiative
Non-Restrictive	User Initiative	Mixed Initiative

Figure 24.10 Operational definition of initiative, following Singh et al. (2002).

Dialogue Management: Confirmation

- Miscommunication common in SDS
  - "Error spirals" of sequential errors
    - Highly problematic
  - Recognition, recovery crucial
- Confirmation strategies can detect, mitigate
  - Explicit confirmation:

Dialogue Management: Confirmation

- Miscommunication common in SDS
  - "Error spirals" of sequential errors
    - Highly problematic
  - Recognition, recovery crucial
- Confirmation strategies can detect, mitigate
  - Explicit confirmation:
    - Ask for verification of each input
  - Implicit confirmation:

Dialogue Management: Confirmation

- Miscommunication common in SDS
  - “Error spirals” of sequential errors
    - Highly problematic
  - Recognition, recovery crucial
- Confirmation strategies can detect, mitigate
  - Explicit confirmation:
    - Ask for verification of each input
  - Implicit confirmation:
    - Include input information in subsequent prompt

Confirmation Strategies

- Explicit:

S: Which city do you want to leave from?
U: Baltimore.
S: **Do you want to leave from Baltimore?**
U: Yes.
U: I'd like to fly from Denver Colorado to New York City on September twenty first in the morning on United Airlines
S: **Let's see then. I have you going from Denver Colorado to New York on September twenty first. Is that correct?**
U: Yes

Confirmation Strategy

- Implicit:

U:	I want to travel to Berlin
S:	**When do you want to travel to Berlin?**
U2:	Hi I'd like to fly to Seattle Tuesday Morning
A3:	**Traveling to Seattle on Tuesday, August eleventh in the morning.** Your full name?

Pros and Cons

- Grounding of user input
  - Weakest grounding
    - I.e. continued att'n, next relevant contibution

Pros and Cons

- Grounding of user input
  - Weakest grounding insufficient
    - I.e. continued att’n, next relevant contibution
  - Explicit:

Pros and Cons

- Grounding of user input
  - Weakest grounding insufficient
    - I.e. continued att'n, next relevant contibution
  - Explicit: highest: repetition
  - Implicit:

Pros and Cons

- Grounding of user input
  - Weakest grounding insufficient
    - I.e. continued att'n, next relevant contibution
  - Explicit: highest: repetition
  - Implicit: demonstration, display
- Explicit;

Pros and Cons

- Grounding of user input
  - Weakest grounding insufficient
    - I.e. continued att'n, next relevant contibution
  - Explicit: highest: repetition
  - Implicit: demonstration, display
- Explicit;
  - Pro: easier to correct; Con: verbose, awkward, non-human
- Implicit:

Pros and Cons

- Grounding of user input
  - Weakest grounding insufficient
    - I.e. continued att'n, next relevant contibution
  - Explicit: highest: repetition
  - Implicit: demonstration, display
- Explicit;
  - Pro: easier to correct; Con: verbose, awkward, non-human
- Implicit:
  - Pro: more natural, efficient; Con: less easy to correct

Rejection

- System recognition confidence is too low
- System needs to reprompt
  - Often repeatedly

Rejection

- System recognition confidence is too low
- System needs to reprompt
  - Often repeatedly
    - Out-of-vocabulary, out-of-grammar inputs
- Strategies: Progressive prompting

Rejection

- System recognition confidence is too low
- System needs to reprompt
  - Often repeatedly
    - Out-of-vocabulary, out-of-grammar inputs
- Strategies: Progressive prompting
  - Initially: ‘rapid reprompting’: ‘What?’, ‘Sorry?’

Rejection

- System recognition confidence is too low
- System needs to reprompt
  - Often repeatedly
    - Out-of-vocabulary, out-of-grammar inputs
- Strategies: Progressive prompting
  - Initially: 'rapid reprompting': 'What?', 'Sorry?'
  - Later: increasing detail

- Progressive prompting

System:	When would you like to leave?
Caller:	Well, um, I need to be in New York in time for the first World Series game.
System:	<reject>. Sorry, I didn't get that. Please say the month and day you'd like to leave.
Caller:	I wanna go on October fifteenth.

VoiceXML

- W3C standard for simple frame-based dialogues
  - Fairly common in commercial settings
- Construct forms, menus
  - Forms get field data
    - Using attached prompts
    - With specified grammar (CFG)
    - With simple semantic attachments

Simple VoiceXML Example

```
<form>
  <field name="transporttype">
     <prompt>
        Please choose airline, hotel, or rental car.
     </prompt>
     <grammar type="application/x=nuance-gsl">
       [airline hotel "rental car"]
     </grammar>
  </field>
  <block>
    <prompt>
     You have chosen <value expr="transporttype">.
    </prompt>
  </block>
</form>
```

Frame-based Systems: Pros and Cons

- Advantages

Frame-based Systems: Pros and Cons

- Advantages
  - Relatively flexible input – multiple inputs, orders
  - Well-suited to complex information access (air)
  - Supports different types of initiative

- Disadvantages

Frame-based Systems: Pros and Cons

- Advantages
  - Relatively flexible input – multiple inputs, orders
  - Well-suited to complex information access (air)
  - Supports different types of initiative

- Disadvantages
  - Ill-suited to more complex problem-solving
    - Form-filling applications

Dialogue Manager Tradeoffs

- Flexibility vs Simplicity/Predictability
  - System vs User vs Mixed Initiative
  - Order of dialogue interaction
  - Conversational "naturalness" vs Accuracy
  - Cost of model construction, generalization, learning, etc

Dialog Systems Design

- User-centered design approach:
  - Study user and task:
    - Interview users; record human-human interactions; systems

Dialog Systems Design

- User-centered design approach:
  - Study user and task:
    - Interview users; record human-human interactions; systems
  - Build simulations and prototypes:
    - Wizard-of-Oz systems (WOZ): Human replaces system
      - Can assess issues in partial system; simulate errors, etc

Dialog Systems Design

- User-centered design approach:
  - Study user and task:
    - Interview users; record human-human interactions; systems
  - Build simulations and prototypes:
    - Wizard-of-Oz systems (WOZ): Human replaces system
      - Can assess issues in partial system; simulate errors, etc
  - Iteratively test on users:
    - Redesign prompts (email subdialog)
    - Identify need for barge-in

SDS Evaluation

- Goal: Determine overall user satisfaction
  - Highlight systems problems; help tune
- Classically: Conduct user surveys

TTS Performance	Was the system easy to understand ?
ASR Performance	Did the system understand what you said?
Task Ease	Was it easy to find the message/flight/train you wanted?
Interaction Pace	Was the pace of interaction with the system appropriate?
User Expertise	Did you know what you could say at each point?
System Response	How often was the system sluggish and slow to reply to you?
Expected Behavior	Did the system work the way you expected it to?
Future Use	Do you think you'd use the system in the future?

Figure 24.14 User satisfaction survey, adapted from Walker et al. (2001).

SDS Evaluation

- User evaluation issues:

SDS Evaluation

- User evaluation issues:
  - Expensive; often unrealistic; hard to get real user to do
- Create model correlated with human satisfaction
- Criteria:

SDS Evaluation

- User evaluation issues:
  - Expensive; often unrealistic; hard to get real user to do
- Create model correlated with human satisfaction
- Criteria:
  - Maximize task success
    - Measure task completion: % subgoals; Kappa of frame values

SDS Evaluation

- User evaluation issues:
  - Expensive; often unrealistic; hard to get real user to do
- Create model correlated with human satisfaction
- Criteria:
  - Maximize task success
    - Measure task completion: % subgoals; Kappa of frame values
  - Minimize task costs
    - Efficiency costs: time elapsed; # turns; # error correction turns
    - Quality costs: # rejections; # barge-in; concept error rate

PARADISE Model

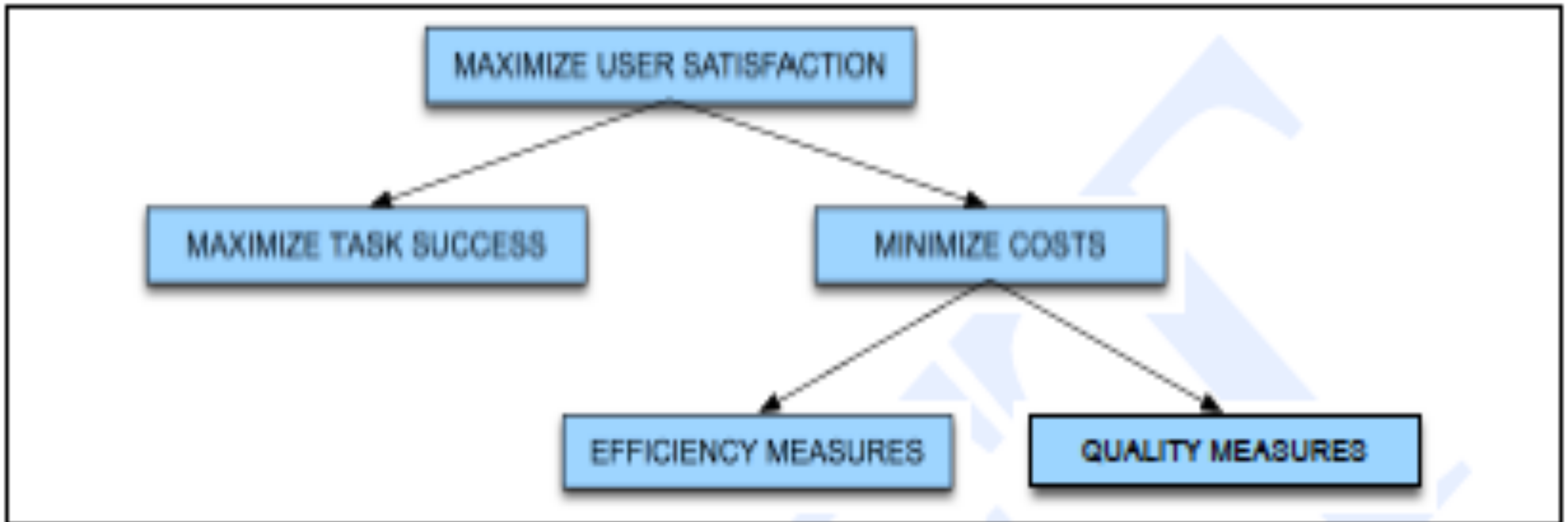


Figure 24.15 PARADISE's structure of objectives for spoken dialogue performance. After Walker et al. (1997).

PARADISE Model

- Compute user satisfaction with questionnaires

PARADISE Model

- Compute user satisfaction with questionnaires
- Extract task success and costs measures from corresponding dialogs
  - Automatically or manually

PARADISE Model

- Compute user satisfaction with questionnaires
- Extract task success and costs measures from corresponding dialogs
  - Automatically or manually
- Perform multiple regression:
  - Assign weights to all factors of contribution to Usat
  - Task success, Concept accuracy key

PARADISE Model

- Compute user satisfaction with questionnaires
- Extract task success and costs measures from corresponding dialogs
  - Automatically or manually
- Perform multiple regression:
  - Assign weights to all factors of contribution to Usat
  - Task success, Concept accuracy key
- Allows prediction of accuracy on new dialog w/Q&A

Information State Dialogue Management

- Problem: Not every task is equivalent to form-filling
- Real tasks require:

Information State Dialogue Management

- Problem: Not every task is equivalent to form-filling
- Real tasks require:
  - Proposing ideas, refinement, rejection, grounding, clarification, elaboration, etc

Information State Dialogue Management

- Problem: Not every task is equivalent to form-filling
- Real tasks require:
  - Proposing ideas, refinement, rejection, grounding, clarification, elaboration, etc
- Information state models include:
  - Information state
  - Dialogue act interpreter
  - Dialogue act generator
  - Update rules
  - Control structure

Information State Systems

- Information state :
  - Discourse context, grounding state, intentions, plans.

Information State Systems

- Information state :
  - Discourse context, grounding state, intentions, plans.
- Dialogue acts:
  - Extension of speech acts, to include grounding acts
    - Request-inform; Confirmation

Information State Systems

- Information state :
  - Discourse context, grounding state, intentions, plans.
- Dialogue acts:
  - Extension of speech acts, to include grounding acts
    - Request-inform; Confirmation
- Update rules
  - Modify information state based on DAs

Information State Systems

- Information state :
  - Discourse context, grounding state, intentions, plans.
- Dialogue acts:
  - Extension of speech acts, to include grounding acts
    - Request-inform; Confirmation
- Update rules
  - Modify information state based on DAs
    - When a question is asked

Information State Systems

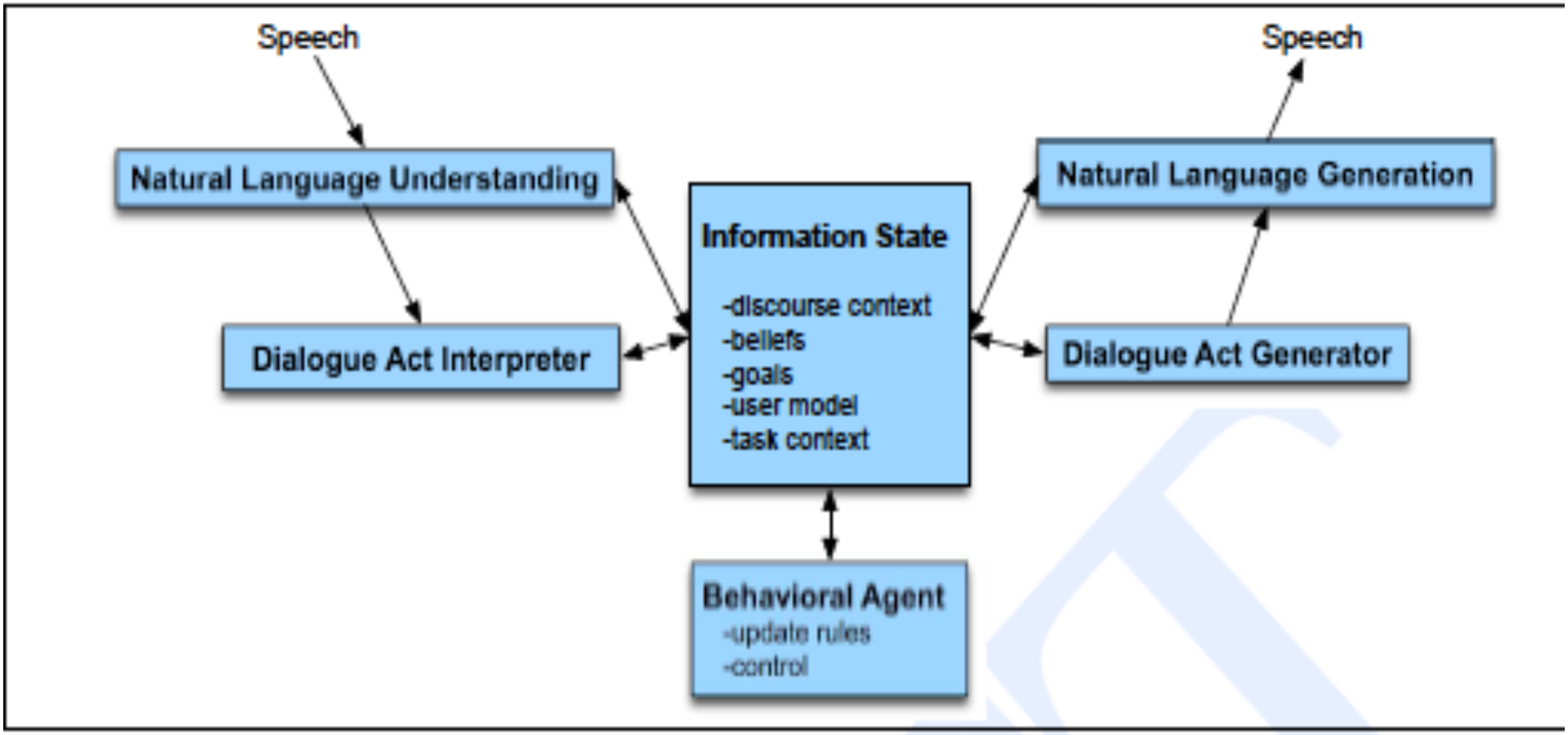
- Information state :
  - Discourse context, grounding state, intentions, plans.
- Dialogue acts:
  - Extension of speech acts, to include grounding acts
    - Request-inform; Confirmation
- Update rules
  - Modify information state based on DAs
    - When a question is asked, answer it
    - When an assertion is made,

Information State Systems

- Information state :
  - Discourse context, grounding state, intentions, plans.
- Dialogue acts:
  - Extension of speech acts, to include grounding acts
    - Request-inform; Confirmation
- Update rules
  - Modify information state based on DAs
    - When a question is asked, answer it
    - When an assertion is made,
      - Add information to context, grounding state

Information State Architecture

- Simple ideas, complex execution



Summary

- Perspectives on the Linguistics of Conversation
  - Differences between general discourse and dialog
    - Turn-taking, speech/dialog acts, structure
    - Implications for spoken dialog systems

- SDS overview
  - Architecture
  - Components & design issues
  - Basic of dialog management
  - Design and evaluation

From Human to Computer

- Conversational agents
  - Systems that (try to) participate in dialogues
  - Examples: Directory assistance, travel info, weather, restaurant and navigation info
- Issues:
  - Limited understanding: ASR errors, interpretation
  - Computational costs:
    - broader coverage -> slower, less accurate

Gesture, Gaze & Voice

- Range of gestural signals:
  - head (nod,shake), shoulder, hand, leg, foot movements; facial expressions; postures; artifacts
  - Align with syllables
- Units: phonemic clause + change
- Study with recorded exchanges

Yielding the Floor

- Turn change signal
  - Offer floor to auditor/hearer
- Cues: pitch fall, lengthening, “but uh”, end gesture, amplitude drop+’uh’, end clause
- Likelihood of change increases with more cues
- Negated by any gesticulation

Taking the Floor

- Speaker-state signal
  - Indicate becoming speaker
- Occurs at beginning of turns
- Cues:
  - Shift in head direction
    - AND/OR
  - Start of gesture

Retaining the Floor

- Within-turn signal
  - Still speaker: Look at hearer as end clause
- Continuation signal
  - Still speaker: Look away after within-turn/back
- Back-channel:
  - ‘mmhm’ /okay/etc; nods,
    - sentence completion. Clarification request; restate
  - NOT a turn: signal attention, agreement, confusion

Segmenting Turns

- Speaker alone:
  - Within-turn signal->end of one unit;
  - Continuation signal -. Beginning of next unit
- Joint signal:
  - Speaker turn signal (end); auditor ->speaker; speaker->auditor
  - Within-turn + back-channel + continuation
    - Back-channels signal understanding
  - Early back-channel + continuation

Regaining Attention

- Gaze & Disfluency
  - Disfluency: “perturbation” in speech
    - Silent pause, filled pause, restart
  - Gaze:
    - Conversants don’t stare at each other constantly
    - However, speaker expects to meet hearer’s gaze
      - Confirm hearer’s attention
- Disfluency occurs when realize hearer NOT attending
  - Pause until begin gazing, or to request attention