SDS Systems & Components

Ling575
Spoken Dialog Systems
April 7, 2016

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?
  - 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

- 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.**

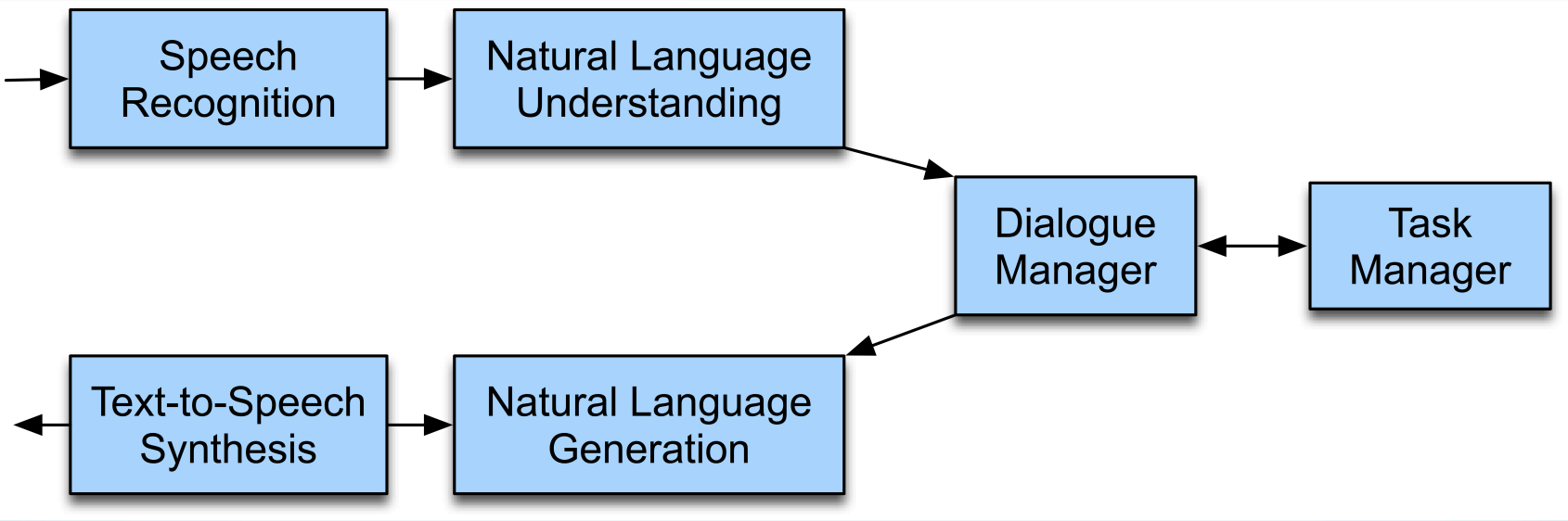
Quantity

- A:How much money do you have on you?
- B: I have 5 dollars
  - Implication: not 6 dollars
- 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

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

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

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

  Alternatives:
  - Full parser with semantic attachments
  - Domain-specific analyzers
- SHOW:
- FLIGHTS:
  - ORIGIN:
    - CITY: Boston
    - DATE:
      - DAY-OF-WEEK: Tuesday
    - TIME:
      - PART-OF-DAY: Morning
  - DEST:
    - CITY: San Francisco

Generation and TTS

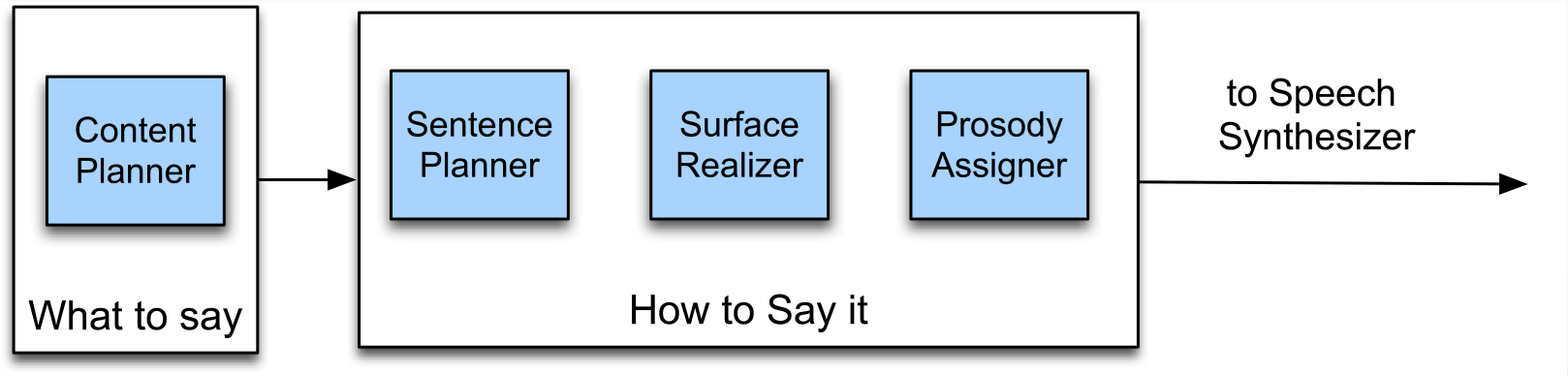
- Generation:
  - Identify concepts to express
  - Convert to words
  - Assign appropriate prosody, intonation
- TTS:
  - Input words, prosodic markup
  - Synthesize acoustic waveform

Generation

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



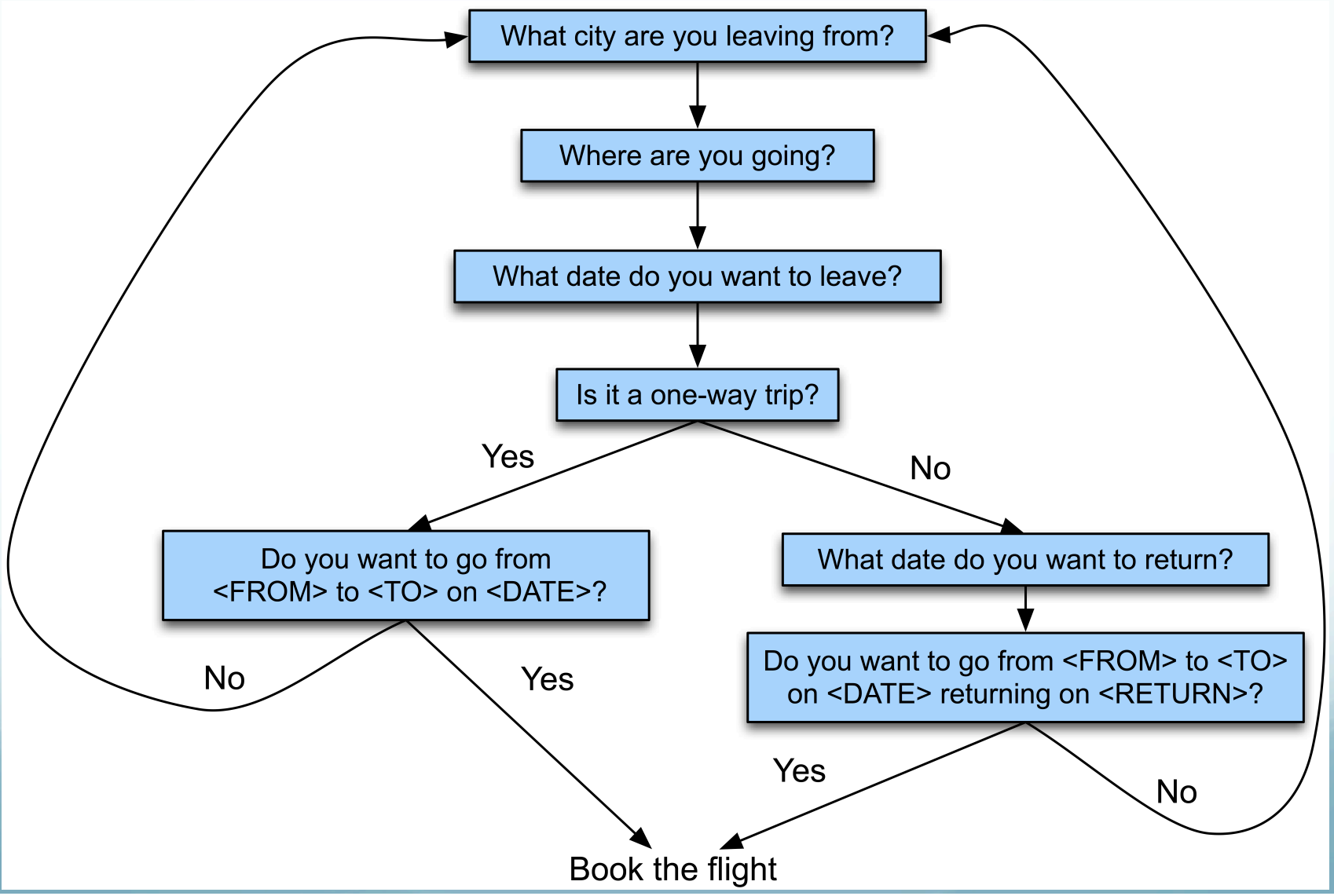
Dialogue Manager

- 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
- Statistical Dialog Management

Finite-State Management



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?
  - 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'

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

- Essentially form-filling
  - User can include any/all of the pieces of form
  - System must determine which entered, remain
  - Rules determine next action, question, information presentation

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?"

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

Dialogue Management: Confirmation

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

Dialog Example

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.

Travel Planning

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*

Figure 19.1 The travel domain: a fragment from a successful conversation between a user (U) and the Communicator system (S) of Xu and Rudnicky (2000).

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 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

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

Richer Dialog Management

- Alternative Dialog Management approaches
  - More flexible interaction, motivated by human-human
  - Information State
    - General interpretation of speech in terms of dialog acts
      - Similar to “speech acts”, e.g. statement, wh-q, yn-q, check,..
    - Model of knowledge, belief state of current dialog
  - Statistical dialog management
    - Builds on reinforcement learning approaches (planning)
    - Aims to automatically learns best sequence of actions
    - Models uncertainty in system understanding of user

Designing Dialog

- Apply user-centered design
  - Study user and task: How?
    - Interview potential users, record human-human tasks
  - Study how the user interacts with the system
    - But it's not built yet....
      - Wizard-of-Oz systems: Simulations
        - User thinks they're interacting with a system, but it's driven by a human
      - Prototypes
  - Iterative redesign:
    - Test system: see how users really react, what problems occur, correct, repeat

SDS Evaluation

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

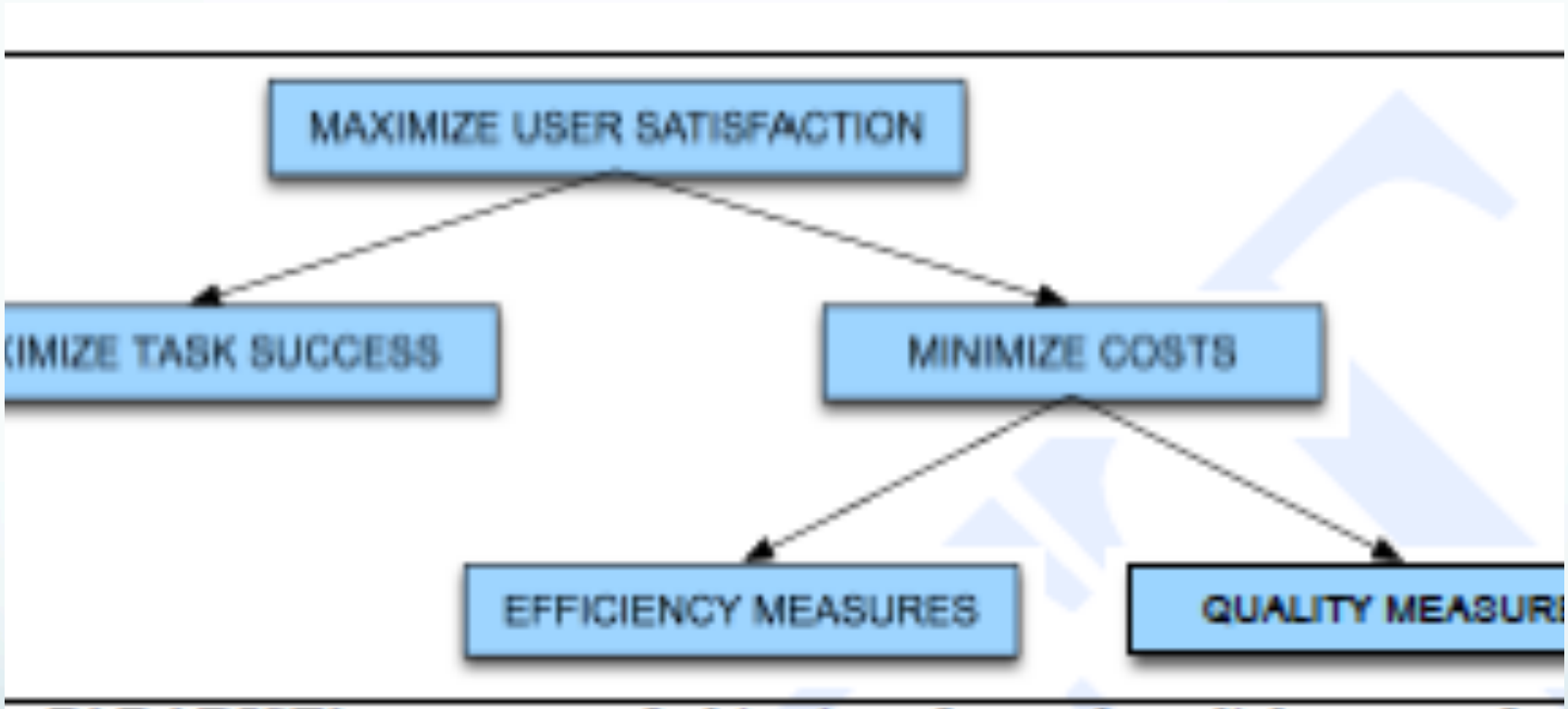
TTS Performance	Was the system easy to understand ?
ASR Performance	Did the system understand what you said?
Task Ease	Was it easy to find the message/flight/train you wanted?
Interaction Pace	Was the pace of interaction with the system appropriate?
User Expertise	Did you know what you could say at each 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:
  - 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



PARADISE's structure of objectives for spoken dialogue perfor
1997).

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

Summary

- Spoken Dialogue Systems:
  - Build on existing text-based NLP techniques, but
  - Incorporate dialogue specific factors:
    - Turn-taking, grounding, dialogue acts
  - Affected by computational and modal constraints
    - Recognition errors, processing speed, etc.
    - Speech transience, slowness
  - Becoming more widespread and more flexible

Components: ASR

Drawing heavily on resource slides from Speech and Language Processing, Jurafsky and Martin

Speech Recognition

- Applications of Speech Recognition (ASR)
  - Dictation
  - Telephone-based Information (directions, air travel, banking, etc)
  - Hands-free (in car)
  - Speaker Identification
  - Language Identification
  - Second language ('L2') (accent reduction)
  - Audio archive searching

LVCSR

- Large Vocabulary Continuous Speech Recognition
- ~20,000-64,000 words
- Speaker independent (vs. speaker-dependent)
- Continuous speech (vs isolated-word)

Current error rates

Ballpark numbers; exact numbers depend very much on the specific corpus

Task	Vocabulary	Error Rate%
Digits	11	0.5
WSJ read speech	20K	3
Broadcast news	64,000+	10
CTS SWBD (GMM) 300hrs	64,000+	23-27
CTS SWBD (DNN) 300hrs	64,000+	16-18
CTS SWBD (GMM) >1000hr	64,000+	17-18
CTS SWBD (DNN) >>1000hr	64,000+	~8
Google Voice > 5800hrs		12
YouTube > 1,400hrs		47

(some results Hinton 2012)

HSR versus ASR

Task	Vocab	ASR	Hum SR
Continuous digits	**11**	**.5**	**.009**
WSJ 1995 clean	**5K**	**3**	**0.9**
WSJ 1995 w/noise	**5K**	**9**	**1.1**
SWBD 2004	**65K**	**~8**	**4**

- Conclusions:
  - Machines about 5 times worse than humans
  - Gap increases with noisy speech
  - These numbers are rough, take with grain of salt