# CS 224S/LING 281 Speech Recognition, Synthesis, and Dialogue

Dan Jurafsky

Lecture 13: Dialogue: Information State Systems and Dialogue Act Interpretation

# Outline

- Natural Language Understanding
- Natural Language Generation
- Information-State Models
  - Dialogue-Act Detection
  - Dialogue-Act Generation

# Summary

- The Linguistics of Conversation
- Basic Conversational Agents
  - ASR
  - NLU
  - Generation
  - Dialogue Manager
- Dialogue Manager Design
  - Finite State
  - Frame-based
  - Initiative: User, System, Mixed
- VoiceXML
- Advanced issues in NLU and Generation
- Information-State
  - Dialogue-Act Detection
  - Dialogue-Act Generation
- Evaluation
- Utility-based conversational agents
  - MDP, POMDP

# Natural Language Understanding

# Semantics for a sentence

LIST FLIGHTS ORIGIN

Show me flights from Boston

DESTINATION DEPARTDATE

to San Francisco on Tuesday

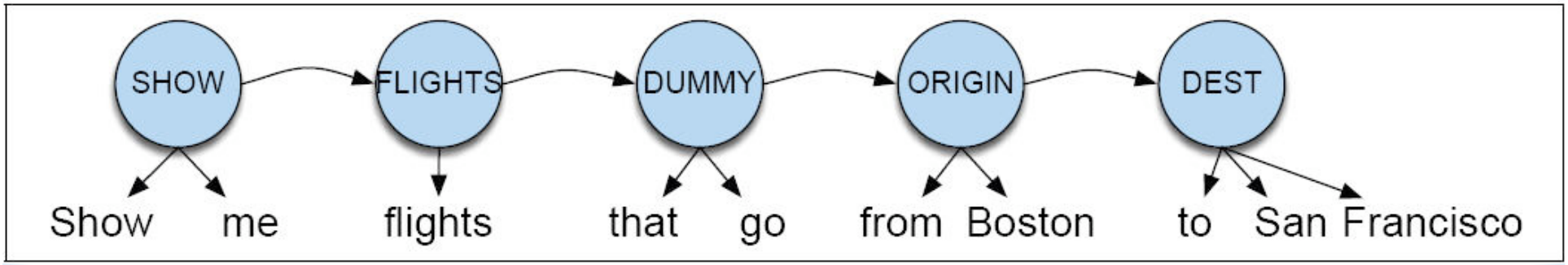
DEPARTTIME

morning

# HMMs for semantics

- Idea: use an HMM for semantics, just as we did for ASR (and part-of-speech tagging, etc)
- Hidden units:
  - Semantic slot names
    - Origin
    - Destination
    - Departure time
- Observations:
  - Word sequences

# HMM model of semantics - Pieraccini et al (1991)



# Semantic HMM

- Goal of HMM model:
  - to compute labeling of semantic roles C = $c_1, c_2, \ldots, c_n$ (C for 'cases' or 'concepts')
  - that is most probable given words W

$$\begin{aligned}\operatorname*{argmax}_C P(C \mid W) &= \operatorname*{argmax}_C \frac{P(W \mid C)P(C)}{P(W)} \\ &= \operatorname*{argmax}_C P(W \mid C)P(C) \\ &= \operatorname*{argmax}_C \prod_{i=2}^{N} P(w_i \mid w_{i-1} \ldots w_1, C) P(w_1 \mid C) \prod_{i=2}^{M} P(c_i \mid c_{i-1} \ldots c_1)\end{aligned}$$

# Semantic HMM

- From previous slide:

$$= \underset{C}{\operatorname{argmax}} \prod_{i=2}^{N} P(w_i \mid w_{i-1} \ldots w_1, C) P(w_1 \mid C) \prod_{i=2}^{M} P(c_i \mid c_{i-1} \ldots c_1)$$

- Assume simplification:

$$P(w_i \mid w_{i-1} \ldots w_1, C) = P(w_i \mid w_{i-1}, \ldots, w_{i-N+1}, c_i)$$

$$P(c_i \mid c_{i-1} \ldots c_1, C) = P(c_i \mid c_{i-1}, \ldots, c_{i-M+1})$$

- Final form:

$$= \underset{C}{\operatorname{argmax}} \prod_{i=2}^{N} P(w_i \mid w_{i-1} \ldots w_{i-N+1}, c_i) \prod_{i=2}^{M} P(c_i \mid c_{i-1} \ldots c_{i-M+1})$$

# Semi-HMMs

- Each hidden state
  - Can generate multiple observations
- By contrast, a traditional HMM
  - One observation per hidden state
  - Need to loop to have multiple observations with the same state label

# Another way to do NLU: Semantic Grammars

- 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

# An example of a frame

- Show me morning flights from Boston to SF on Tuesday.

```
SHOW:
   FLIGHTS:
     ORIGIN:
          CITY: Boston
          DATE: Tuesday
          TIME: morning
     DEST:
          CITY: San Francisco
```

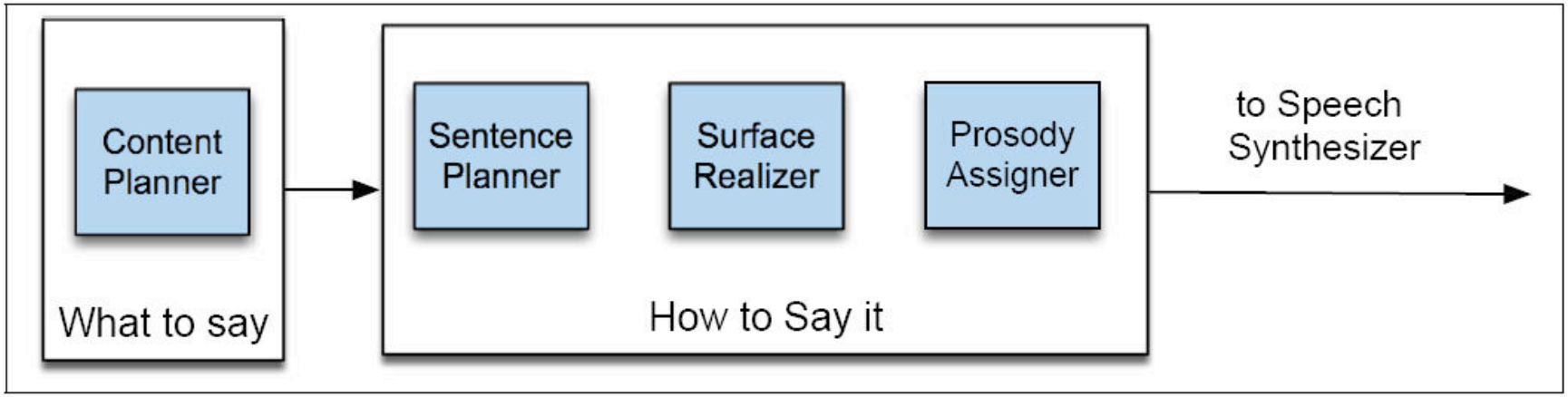
# Generation Component

- Content Planner
  - Decides what content to express to user
    - (ask a question, present an answer, etc)
  - Often merged with dialogue manager
- Language Generation
  - Chooses syntactic structures and words to express meaning.
  - Simplest method
    - All words in sentence are prespecified!
    - "Template-based generation"
    - Can have variables:
      - What time do you want to leave CITY-ORIG?
      - Will you return to CITY-ORIG from CITY-DEST?

# More sophisticated language generation component

- Natural Language Generation
- This is a field, like Parsing, or Natural Language Understanding, or Speech Synthesis, with its own (small) conference
- Approach:
  - Dialogue manager builds representation of meaning of utterance to be expressed
  - Passes this to a "generator"
  - Generators have three components
    - Sentence planner
    - Surface realizer
    - Prosody assigner

# Architecture of a generator for a dialogue system
## (after Walker and Rambow 2002)



# HCI constraints on generation for dialogue: "Coherence"

- Discourse markers and pronouns ("Coherence"):

(1) Please say the date.

…Please say the start time.

…Please say the duration…

…Please say the subject…

(2) First, tell me the date.

…Next, I'll need the time it starts.

…Thanks. <pause> Now, how long is it supposed to last?

…Last of all, I just need a brief description

**Bad!**

**Good!**

# HCI constraints on generation for dialogue: coherence (II): tapered prompts

- Prompts which get incrementally shorter:
- System: Now, what's the first company to add to your watch list?
- Caller: Cisco
- System: What's the next company name? (Or, you can say, "Finished")
- Caller: IBM
- System: Tell me the next company name, or say, "Finished."
- Caller: Intel
- System: Next one?
- Caller: America Online.
- System: Next?
- Caller: …

# How mixed initiative is usually defined

- First we need to define two other factors
- Open prompts vs. directive prompts
- Restrictive versus non-restrictive grammar

# Open vs. Directive Prompts

- Open prompt
  - System gives user very few constraints
  - User can respond how they please:
  - "How may I help you?" "How may I direct your call?"
- Directive prompt
  - Explicit instructs user how to respond
  - "Say yes if you accept the call; otherwise, say no"

# Restrictive vs. Non-restrictive grammars

- Restrictive grammar
  - Language model which strongly constrains the ASR system, based on dialogue state
- Non-restrictive grammar
  - Open language model which is not restricted to a particular dialogue state

# Definition of Mixed Initiative

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

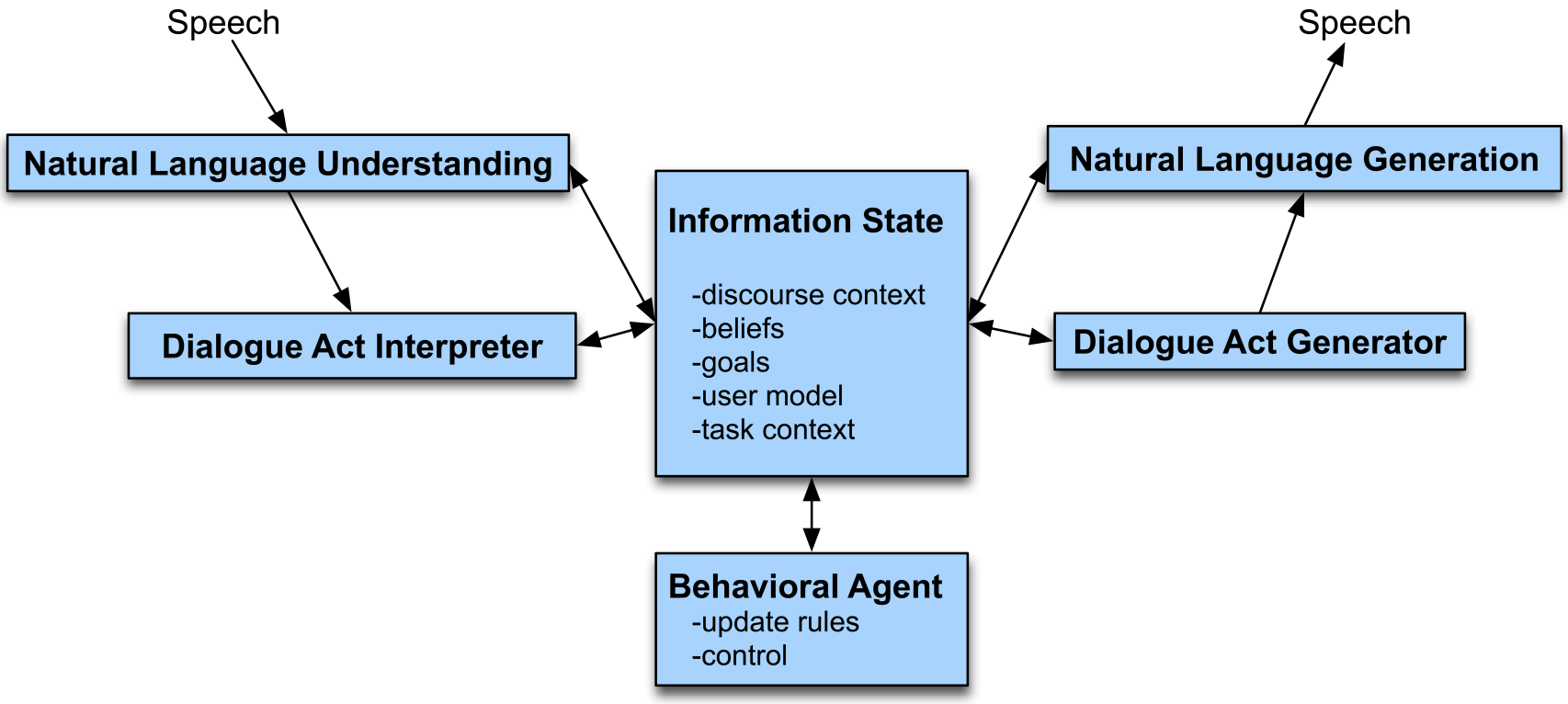
# Information-State and Dialogue Acts

- If we want a dialogue system to be more than just form-filling
- Needs to:
  - Decide when the user has asked a question, made a proposal, rejected a suggestion
  - Ground a user's utterance, ask clarification questions, suggestion plans
- Suggests:
  - Conversational agent needs sophisticated models of interpretation and generation
    - In terms of speech acts and grounding
    - Needs more sophisticated representation of dialogue context than just a list of slots

# Information-state architecture

- Information state
- Dialogue act interpreter
- Dialogue act generator
- Set of update rules
  - Update dialogue state as acts are interpreted
  - Generate dialogue acts
- Control structure to select which update rules to apply

# Information-state



# Dialogue acts

- Also called "conversational moves"
- An act with (internal) structure related specifically to its dialogue function
- Incorporates ideas of grounding
- Incorporates other dialogue and conversational functions that Austin and Searle didn't seem interested in

# 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 |
| FEEDBACK | Okay |
| DELIBERATE | Let me check my calendar here |
| CONFIRM | Okay, that would be wonderful |
| CLARIFY | Okay, do you mean Tuesday the 23rd? |

# DAMSL: forward looking func.

| | |
|---|---|
| STATEMENT | a claim made by the speaker |
| INFO-REQUEST | a question by the speaker |
| CHECK | a question for confirming information |
| INFLUENCE-ON-ADDRESSEE | (=Searle's directives) |
| OPEN-OPTION | a weak suggestion or listing of options |
| ACTION-DIRECTIVE | an actual command |
| INFLUENCE-ON-SPEAKER | (=Austin's commissives) |
| OFFER | speaker offers to do something |
| COMMIT | speaker is committed to doing something |
| CONVENTIONAL | other |
| OPENING | greetings |
| CLOSING | farewells |
| THANKING | thanking and responding to thanks |

# DAMSL: backward looking func.

AGREEMENT speaker's response to previous proposal
- ACCEPT accepting the proposal
- ACCEPT-PART accepting some part of the proposal
- MAYBE neither accepting nor rejecting the proposal
- REJECT-PART rejecting some part of the proposal
- REJECT rejecting the proposal
- HOLD putting off response, usually via subdialogue

ANSWER answering a question

UNDERSTANDING whether speaker understood previous
- SIGNAL-NON-UNDER. speaker didn't understand
- SIGNAL-UNDER. speaker did understand
  - ACK demonstrated via continuer or assessment
  - REPEAT-REPHRASE demonstrated via repetition or reformulation
  - COMPLETION demonstrated via collaborative completion

# A DAMSL Labeling

| | | |
|---|---|---|
| [assert] | $C_1$: | . . . I need to travel in May. |
| [info-req,ack] | $A_1$: | And, what day in May did you want to travel? |
| [assert, answer] | $C_2$: | OK uh I need to be there for a meeting that's from the 12th to the 15th. |
| [info-req,ack] | $A_2$: | And you're flying into what city? |
| [assert,answer] | $C_3$: | Seattle. |
| [info-req,ack] | $A_3$: | And what time would you like to leave Pittsburgh? |
| [check,hold] | $C_4$: | Uh hmm I don't think there's many options for non-stop. |
| [accept,ack] | $A_4$: | Right. |
| [assert] | | There's three non-stops today. |
| [info-req] | $C_5$: | What are they? |
| [assert, open-option] | $A_5$: | 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. |
| [accept,ack] | $C_6$: | OK I'll take the 5ish flight on the night before on the 11th. |
| [check,ack] | $A_6$: | On the 11th? |
| [assert,ack] | | OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115. |
| [ack] | $C_7$: | OK. |

# Conversation Acts
## Traum and Hinkelman (1992)

| Act Type | Sample Acts |
|---|---|
| turn-taking | take-turn, keep-turn, release-turn, assign-turn |
| grounding | acknowledge, repair, continue |
| core speech acts | inform, wh-question, accept, request, offer |
| argumentation | elaborate, summarize, question-answer, clarify |

# Automatic Interpretation of Dialogue Acts

- How do we automatically identify dialogue acts?
- Given an utterance:
  - Decide whether it is a QUESTION, STATEMENT, SUGGEST, or ACK
- Recognizing illocutionary force will be crucial to building a dialogue agent
- Perhaps we can just look at the form of the utterance to decide?

# Can we just use the surface syntactic form?

- YES-NO-Q's have auxiliary-before-subject syntax:
  - Will breakfast be served on USAir 1557?
- STATEMENTs have declarative syntax:
  - I don't care about lunch
- COMMAND's have imperative syntax:
  - Show me flights from Milwaukee to Orlando on Thursday night

# Surface form != speech act type

| | Locutionary Force | Illocutionary Force |
|---|---|---|
| Can I have the rest of your sandwich? | Question | Request |
| I want the rest of your sandwich | Declarative | Request |
| Give me your sandwich! | Imperative | Request |

# Dialogue act disambiguation is hard! Who's on First?

Abbott: Well, Costello, I'm going to New York with you. Bucky Harris the Yankee's manager gave me a job as coach for as long as you're on the team.
Costello: Look Abbott, if you're the coach, you must know all the players.
Abbott: I certainly do.
Costello: Well you know I've never met the guys. So you'll have to tell me their names, and then I'll know who's playing on the team.
Abbott: Oh, I'll tell you their names, but you know it seems to me they give these ball players now-a-days very peculiar names.
Costello: You mean funny names?
Abbott: Strange names, pet names...like Dizzy Dean...
Costello: His brother Daffy Abbott: Daffy Dean...
Costello: And their French cousin.
Abbott: French?
Costello: Goofe'
Abbott: Goofe' Dean. Well, let's see, we have on the bags, Who's on first, What's on second, I Don't Know is on third...
Costello: That's what I want to find out.
Abbott: I say Who's on first, What's on second, I Don't Know's on third.

# Dialogue act ambiguity

- Who's on first?
  - INFO-REQUEST
  - or
  - STATEMENT

# Dialogue Act ambiguity

- Can you give me a list of the flights from Atlanta to Boston?
  - This looks like an INFO-REQUEST.
  - If so, the answer is:
    - YES.
  - But really it's a DIRECTIVE or REQUEST, a polite form of:
  - Please give me a list of the flights…
- What looks like a QUESTION can be a REQUEST

# Dialogue Act ambiguity

- Similarly, what looks like a STATEMENT can be a QUESTION:

| | | |
|---|---|---|
| Us | OPEN-OPTION | **I was wanting to make some arrangements for a trip that I'm going to be taking uh to LA uh beginnning of the week after next** |
| Ag | HOLD | **OK uh let me pull up your profile and I'll be right with you here. [pause]** |
| Ag | CHECK | And you said you wanted to travel next week? |
| Us | ACCEPT | **Uh yes.** |

# Indirect speech acts

- Utterances which use a surface statement to ask a question
- Utterances which use a surface question to issue a request

# DA interpretation as statistical classification

- Lots of clues in each sentence that can tell us which DA it is:
- Words and Collocations:
  - *Please* or *would you*: good cue for REQUEST
  - *Are you*: good cue for INFO-REQUEST
- Prosody:
  - Rising pitch is a good cue for INFO-REQUEST
  - Loudness/stress can help distinguish *yeah*/AGREEMENT from *yeah*/BACKCHANNEL
- Conversational Structure
  - *Yeah* following a proposal is probably AGREEMENT; *yeah* following an INFORM probably a BACKCHANNEL

# Statistical classifier model of dialogue act interpretation

- Our goal is to decide for each sentence what dialogue act it is
- This is a **classification task** (we are making a 1-of-N classification decision for each sentence)
- With **N** classes (= number of dialog acts).
- Three probabilistic models corresponding to the 3 kinds of cues from the input sentence.
  - Conversational Structure: Probability of one dialogue act following another P(Answer|Question)
  - Words and Syntax: Probability of a sequence of words given a dialogue act: P("do you" | Question)
  - Prosody: probability of prosodic features given a dialogue act : P("rise at end of sentence" | Question)

# An example of dialogue act detection: Correction Detection

- Despite all these clever confirmation/rejection strategies, dialogue systems still make mistakes (Surprise!)
- If system misrecognizes an utterance, and either
  - Rejects
  - Via confirmation, displays its misunderstanding
- Then user has a chance to make a correction
  - Repeat themselves
  - Rephrasing
  - Saying "no" to the confirmation question.

# Corrections

- Unfortunately, corrections are harder to recognize than normal sentences!
  - Swerts et al (2000): corrections misrecognized twice as often (in terms of WER) as non-corrections!!!
  - Why?
    - Prosody seems to be largest factor: hyperarticulation
    - English Example from Liz Shriberg
      - "NO, I am DE-PAR-TING from Jacksonville)
    - A German example from Bettina Braun from a talking elevator

# Corrections

- Unfortunately, corrections are harder to recognize than normal sentences!
  - Swerts et al (2000): corrections misrecognized twice as often (in terms of WER) as non-corrections!!!
  - Why?
    - Prosody seems to be largest factor: hyperarticulation
    - English Example from Liz Shriberg
      - "NO, I am DE-PAR-TING from Jacksonville"
    - A German example from Bettina Braun from a talking elevator
      - "In den VIERTEN Stock"

# A Labeled dialogue (Swerts et al)

| | Turn | Corr Type |
|---|---|---|
| S | On which day of the week do you want to leave? | |
| S | Sorry, I didn't hear anything from you. Please tell me your departure date | |
| U | *Um tomorrow* | NA |
| S | You can say the departure date. For example, you can say 'today' or 'tomorrow'. Or you can say 'Monday', 'Tuesday', 'Wednesday'... | |
| U | *Tomorrow* | REP |
| S | You can say the departure date. For example, you can say 'today' or 'tomorrow'. Or you can say 'Monday', 'Tuesday', 'Wednesday'... | |
| U | *Saturday* | PAR |
| S | Do you want to leave on Saturday? | |
| U | Yes | |
| S | What time do you want to leave? | |
| U | Eight o'clock A M | |
| S | Do you want to leave around ten oclock p-m | |
| U | *Eight o'clock* | OMIT |
| S | Do you want to leave around ten oclock p-m | |
| U | *No* | OMIT |
| S | What time do you want to leave? | |
| U | *Eight o'clock A M* | ADD |

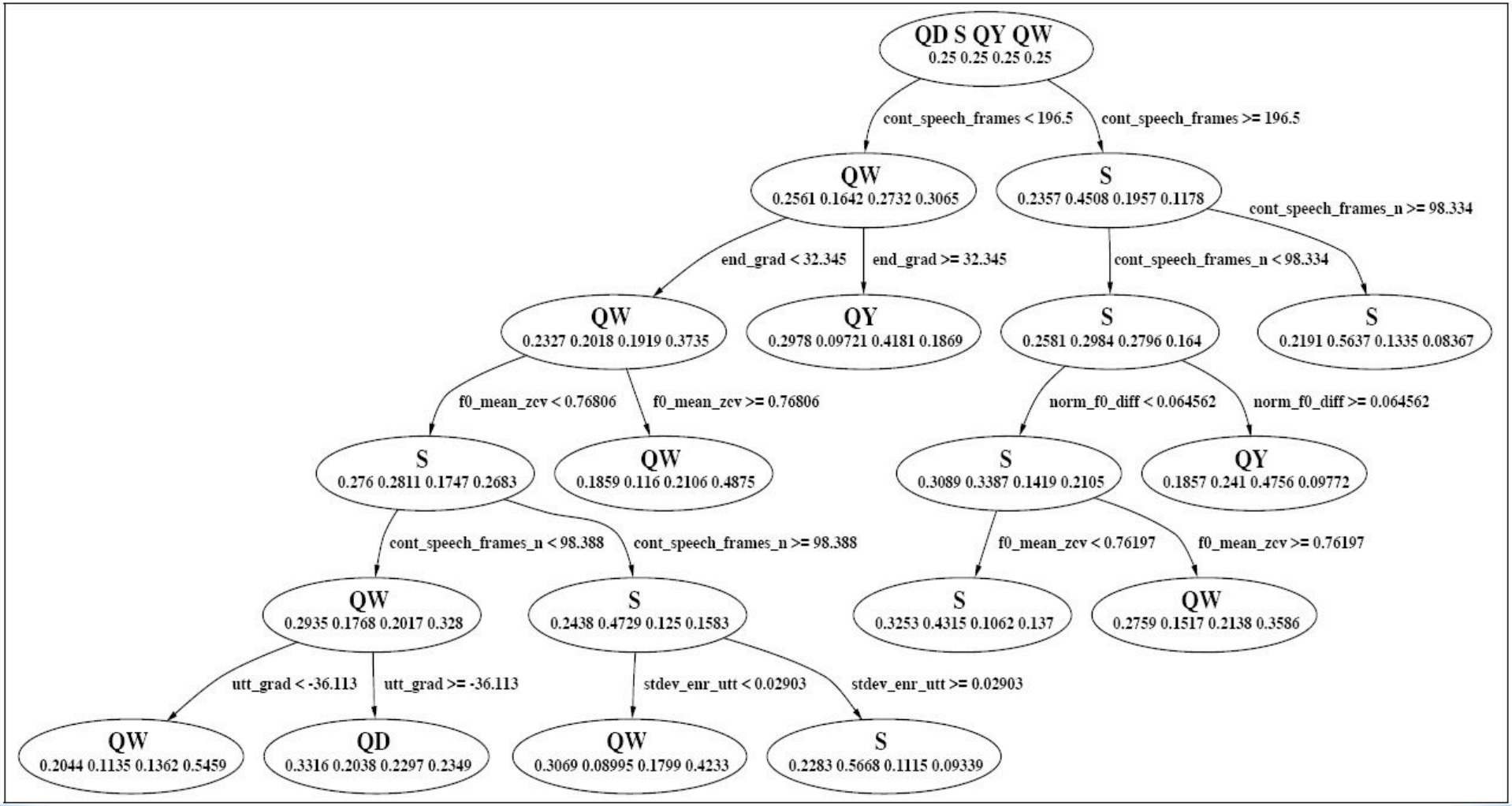
# Machine learning to detect user corrections

- Build classifiers using features like
  - Lexical information (words "no", "correction", "I don't", swear words)
  - Prosodic features (various increases in F0 range, pause duration, and word duration that correlation with hyperarticulation)
  - Length
  - ASR confidence
  - LM probability
  - Various dialogue features (repetition)

# Prosodic Features

- Shriberg et al. (1998)
- Decision tree trained on simple acoustically-based prosodic features
  - Slope of F0 at the end of the utterance
  - Average energy at different places in utterance
  - Various duration measures
  - All normalized in various ways
- These helped distinguish
  - Statement (S)
  - Yes-no-question (QY)
  - Declarative question (QD) ("You're going to the store?")
  - Wh-question (QW)

# Prosodic Decision Tree for making S/ QY/QW/QD decision



# Generating Dialogue Acts

- Confirmation
- Rejection

# Confirmation

- Another reason for grounding
- Errors: Speech is a pretty errorful channel
  - Even for humans; so they use grounding to confirm that they heard correctly
- ASR is way worse than humans!
- So dialogue systems need to do even more grounding and confirmation than humans

# Explicit confirmation

- S: Which city do you want to leave from?
- U: Baltimore
- S: Do you want to leave from Baltimore?
- U: Yes

# 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?

# Implicit vs. Explicit

- Complementary strengths
- Explicit: easier for users to correct systems's mistakes (can just say "no")
- But explicit is cumbersome and long
- Implicit: much more natural, quicker, simpler (if system guesses right).

# Implicit and Explicit

- Early systems: all-implicit or all-explicit
- Modern systems: adaptive
- How to decide?
  - ASR system can give confidence metric.
  - This expresses how convinced system is of its transcription of the speech
  - If high confidence, use implicit confirmation
  - If low confidence, use explicit confirmation

# Computing confidence

- Simplest: use acoustic log-likelihood of user's utterance
- More features
  - Prosodic: utterances with longer pauses, F0 excursions, longer durations
  - Backoff: did we have to backoff in the LM?
  - Cost of an error: Explicit confirmation before moving money or booking flights

# Rejection

- e.g., VoiceXML “nomatch”
- “I’m sorry, I didn’t understand that.”
- Reject when:
  - ASR confidence is low
  - Best interpretation is semantically ill-formed
- Might have four-tiered level of confidence:
  - Below confidence threshhold, reject
  - Above threshold, explicit confirmation
  - If even higher, implicit confirmation
  - Even higher, no confirmation

# Dialogue System Evaluation

- Key point about SLP.
- Whenever we design a new algorithm or build a new application, need to evaluate it
- Two kinds of evaluation
  - **Extrinsic:** embedded in some external task
  - **Intrinsic:** some sort of more local evaluation.

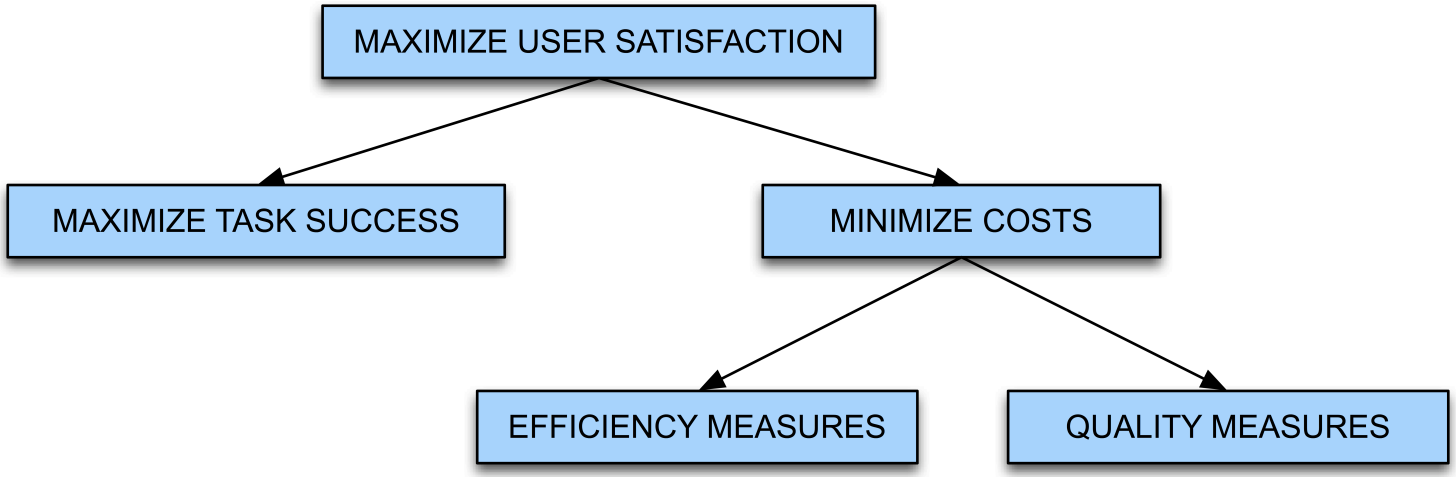
- How to evaluate a dialogue system?
- What constitutes success or failure for a dialogue system?

# Dialogue System Evaluation

- It turns out we'll need an evaluation metric for two reasons
  - 1) the normal reason: we need a metric to help us compare different implementations
    - can't improve it if we don't know where it fails
    - Can't decide between two algorithms without a goodness metric
  - 2) a new reason: we will need a metric for "how good a dialogue went" as an input to reinforcement learning:
    - automatically improve our conversational agent performance via learning

# PARADISE evaluation

- Maximize Task Success
- Minimize Costs
  - Efficiency Measures
  - Quality Measures
- PARADISE (PARAdigm for Dialogue System Evaluation) (Walker *et al*. 2000)



# Task Success

- % of subtasks completed
- Correctness of each questions/answer/ error msg
- Correctness of total solution
  - Attribute-Value matrix (AVM)
  - Kappa coefficient
- Users' perception of whether task was completed

# Task Success

- **Task goals seen as Attribute-Value Matrix**

  *ELVIS e-mail retrieval task* ***(Walker et al '97)***

  ***"Find the time and place of your meeting with Kim."***

| ***Attribute*** | ***Value*** |
|---|---|
| ***Selection Criterion*** | ***Kim or Meeting*** |
| ***Time*** | ***10:30 a.m.*** |
| ***Place*** | ***2D516*** |

- **Task success can be defined by match between AVM values at end of task with "true" values for AVM**

Slide from Julia Hirschberg

# Efficiency Cost

- Polifroni et al. (1992), Danieli and Gerbino (1995) Hirschman and Pao (1993)

- Total elapsed time in seconds or turns
- Number of queries
- Turn correction ration: number of system or user turns used solely to correct errors, divided by total number of turns

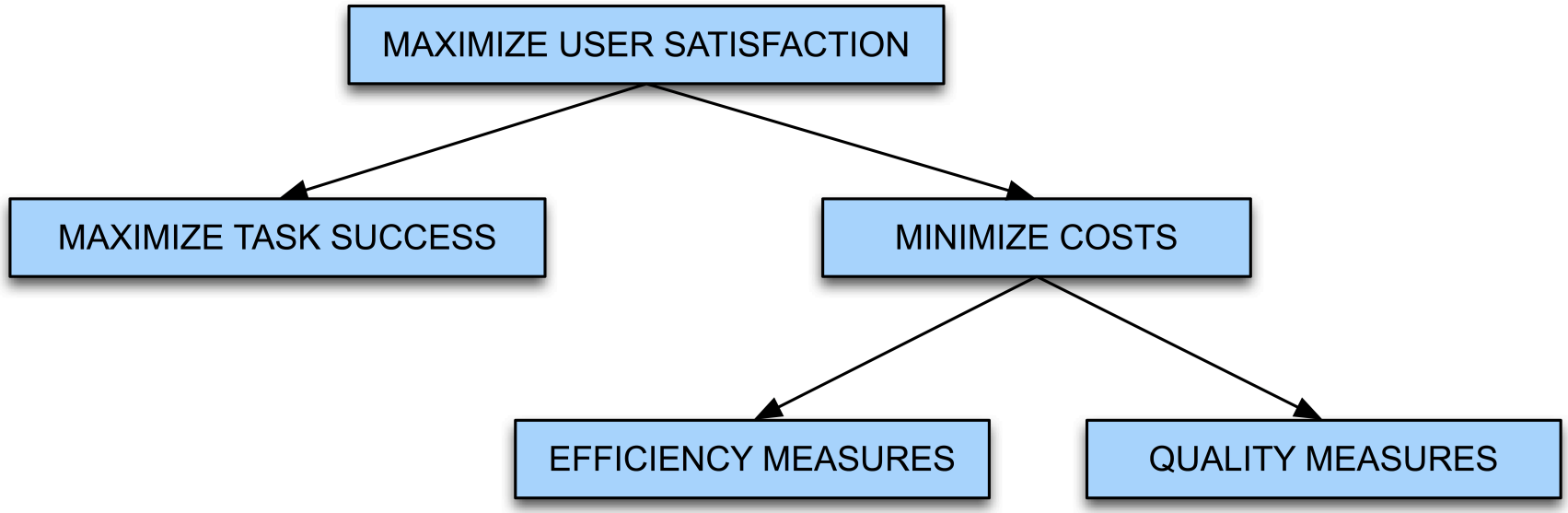
# Quality Cost

- # of times ASR system failed to return any sentence
- # of ASR rejection prompts
- # of times user had to barge-in
- # of time-out prompts
- Inappropriateness (verbose, ambiguous) of system's questions, answers, error messages

# Another key quality cost

- "Concept accuracy" or "Concept error rate"
- % of semantic concepts that the NLU component returns correctly
- I want to arrive in Austin at 5:00
  - DESTCITY: Boston
  - Time: 5:00
- Concept accuracy = 50%
- Average this across entire dialogue
- "How many of the sentences did the system understand correctly"

# PARADISE: Regress against user satisfaction



# Regressing against user satisfaction

- Questionnaire to assign each dialogue a "user satisfaction rating": this is dependent measure
- Set of cost and success factors are independent measures
- Use regression to train weights for each factor

# Experimental Procedures

- Subjects given specified tasks
- Spoken dialogues recorded
- Cost factors, states, dialog acts automatically logged; ASR accuracy,barge-in hand-labeled
- Users specify task solution via web page
- Users complete User Satisfaction surveys
- Use multiple linear regression to model User Satisfaction as a function of Task Success and Costs; test for significant predictive factors

Slide from Julia Hirschberg

# User Satisfaction: Sum of Many Measures

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

# Performance Functions from Three Systems

- ELVIS User Sat.= .21* COMP + .47 * MRS - .15 * ET
- TOOT User Sat.= .35* COMP + .45* MRS - .14*ET
- ANNIE User Sat.= .33*COMP + .25* MRS +.33* Help
  - COMP: User perception of task completion (task success)
  - MRS: Mean (concept) recognition accuracy (cost)
  - ET: Elapsed time (cost)
  - Help: Help requests (cost)

Slide from Julia Hirschberg

# Performance Model

- Perceived task completion and mean recognition score (concept accuracy) are consistently significant predictors of User Satisfaction
- Performance model useful for system development
  - Making predictions about system modifications
  - Distinguishing 'good' dialogues from 'bad' dialogues
  - As part of a learning model

# Now that we have a success metric

- Could we use it to help drive learning?
- In recent work we use this metric to help us learn an optimal policy or strategy for how the conversational agent should behave

# New Idea: Modeling a dialogue system as a probabilistic agent

- A conversational agent can be characterized by:
  - The current knowledge of the system
    - A set of states S the agent can be in
  - a set of actions A the agent can take
  - A goal G, which implies
    - A success metric that tells us how well the agent achieved its goal
    - A way of using this metric to create a strategy or policy $\pi$ for what action to take in any particular state.

# What do we mean by actions A and policies $\pi$?

- Kinds of decisions a conversational agent needs to make:
  - When should I ground/confirm/reject/ask for clarification on what the user just said?
  - When should I ask a directive prompt, when an open prompt?
  - When should I use user, system, or mixed initiative?

# A threshold is a human-designed policy!

- Could we learn what the right action is
  - Rejection
  - Explicit confirmation
  - Implicit confirmation
  - No confirmation
- By learning a policy which,
  - given various information about the current state,
  - dynamically chooses the action which maximizes dialogue success

# Another strategy decision

- Open versus directive prompts
- When to do mixed initiative

- How we do this optimization?
- Markov Decision Processes

# Summary

- The Linguistics of Conversation
- Basic Conversational Agents
  - ASR
  - NLU
  - Generation
  - Dialogue Manager
- Dialogue Manager Design
  - Finite State
  - Frame-based
  - Initiative: User, System, Mixed
- VoiceXML
- Information-State
  - Dialogue-Act Detection
  - Dialogue-Act Generation
- Evaluation
- Utility-based conversational agents
  - MDP, POMDP