

CS 224S LING 281 Speech Recognition and Synthesis

Lecture 14: Dialogue and Conversational Agents (II)
Dan Jurafsky

CS 224S W2006

Outline

- 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

CS 224S W2006

VoiceXML

- Voice eXtensible Markup Language
- An XML-based dialogue design language
- Makes use of ASR and TTS
- Deals well with simple, frame-based mixed initiative dialogue.
- Most common in commercial world (too limited for research systems)
- But useful to get a handle on the concepts.

CS 224S W2006

Voice XML

- Each dialogue is a <form>. (Form is the VoiceXML word for frame)
- Each <form> generally consists of a sequence of <field>s, with other commands

CS 224S W2006

Sample vxml doc

```
<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>
```

CS 224S W2006

VoiceXML interpreter

- Walks through a VXML form in document order
- Iteratively selecting each item
- If multiple fields, visit each one in order.
- Special commands for events

CS 224S W2006

Another vxml doc (1)

```
<noinput>
I'm sorry, I didn't hear you. <reprompt/>
</noinput>
- "noinput" means silence exceeds a timeout threshold

<nomatch>
I'm sorry, I didn't understand that. <reprompt/>
</nomatch>

- "nomatch" means confidence value for utterance is too low
- notice "reprompt" command
```

CS 224S W2006

Another vxml doc (2)

```
<form>
<block> Welcome to the air travel consultant. </block>
<field name="origin">
  <prompt> Which city do you want to leave from? </prompt>
  <grammar type="application/x=nuance-gsl">
    [(san francisco) denver (new york) barcelona]
  </grammar>
  <filled>
    <prompt> OK, from <value expr="origin"> </prompt>
  </filled>
</field>
```

- "filled" tag is executed by interpreter as soon as field filled by user

CS 224S W2006

Another vxml doc (3)

```
<field name="destination">
  <prompt> And which city do you want to go to? </prompt>
  <grammar type="application/x=nuance-gsl">
    [(san francisco) denver (new york) barcelona]
  </grammar>
  <filled>
    <prompt> OK, to <value expr="destination"> </prompt>
  </filled>
</field>
<field name="departdate" type="date">
  <prompt> And what date do you want to leave? </prompt>
  <filled>
    <prompt> OK, on <value expr="departdate"> </prompt>
  </filled>
</field>
```

CS 224S W2006

Another vxml doc (4)

```
<block>
  <prompt> OK, I have you are departing from
    <value expr="origin"> to <value
  expr="destination"> on <value
  expr="departdate">
  </prompt>
  send the info to book a flight...
</block>
</form>
```

CS 224S W2006

Summary: VoiceXML

- Voice eXtensible Markup Language
- An XML-based dialogue design language
- Makes use of ASR and TTS
- Deals well with simple, frame-based mixed initiative dialogue.
- Most common in commercial world (too limited for research systems)
- But useful to get a handle on the concepts.

CS 224S W2006

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

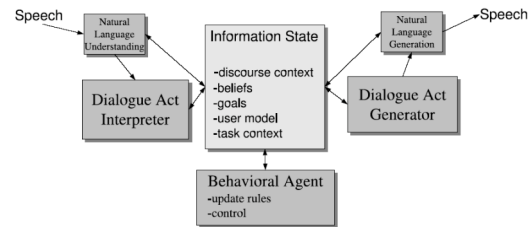
CS 224S W2006

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

CS 224B W2006 14

Information-state



CS 224B W2006 14

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

CS 224B W2006 15

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)

CS 224B W2006 16

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?

CS 224B W2006 17

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

CS 224B W2006 18

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

CS 224S W2006 19

```
[assert] C1: ... I need to travel in May.
[info-req,ack] A1: And, what day in May did you want to travel?
[assert, answer] C2: OK uh I need to be there for a meeting that's from the 12th
to the 15th.
[info-req,ack] A2: And you're flying into what city?
[assert,answer] C3: Seattle.
[info-req,ack] A3: And what time would you like to leave Pittsburgh?
[check,hold] C4: Uh hmm I don't think there's many options for non-stop.
[accept,ack] A4: Right.
[assert] There's three non-stops today.
[info-req] C5: What are they?
[assert, open-option] A5: 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] C6: OK I'll take the 5ish flight on the night before on the 11th.
[check,ack] A6: On the 11th?
[assert,ack] C7: OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air
flight 115.
[ack] C7: OK.
```

CS 224S W2006 20

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?

CS 224S W2006 21

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 Thursday night

CS 224S W2006 22

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

CS 224S W2006 23

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.

CS 224S W2006 24

Dialogue act ambiguity

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

CS 224S W2006 24

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

CS 224S W2006 25

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 beginning 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.

CS 224S W2006 27

Indirect speech acts

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

CS 224S W2006 28

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

CS 224S W2006 29

HMM model of dialogue act interpretation

- A dialogue is an HMM
- The hidden states are the dialogue acts
- The observation sequences are sentences
 - Each observation is one sentence
 - Including words and acoustics
- The observation likelihood model includes
 - N-grams for words
 - Another classifier for prosodic cues
- Summary: 3 probabilistic models:
 - A: Conversational Structure: Probability of one dialogue act following another $P(\text{Answer}|\text{Question})$
 - B: Words and Syntax: Probability of a sequence of words given a dialogue act: $P(\text{"do you"}|\text{Question})$
 - C: Prosody: probability of prosodic features given a dialogue act: $P(\text{"rise at end of sentence"}|\text{Question})$

CS 224S W2006 30

HMMs for dialogue act interpretation

- Goal of HMM model:
 - to compute labeling of dialogue acts $D = d_1, d_2, \dots, d_n$
 - that is most probable given evidence E

$$D^* = \operatorname{argmax}_D P(D|E) = \operatorname{argmax}_D \frac{P(W|D)P(E)}{P(E)}$$

$$= \operatorname{argmax}_D P(E|D)P(D)$$

CS 224S W2006 31

HMMs for dialogue act interpretation

- $D^* = \operatorname{argmax}_D P(E|D)P(D)$
- Let W be word sequence in sentence and F be prosodic feature sequence
- Simplifying (wrong) independence assumption
 - $P(E|D) = P(F|D)P(W|D)$
 - (What are implications of this?)

$$D^* = \operatorname{argmax}_D P(D)P(F|D)P(W|D)$$

CS 224S W2006 32

HMM model for dialogue

$$D^* = \operatorname{argmax}_D P(D)P(F|D)P(W|D)$$

- Three components
 - $P(D)$: probability of sequence of dialogue acts
 - $P(F|D)$: probability of prosodic sequence given one dialogue act
 - $P(W|D)$: probability of word string in a sentence given dialogue act

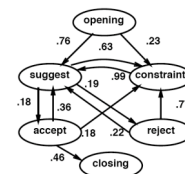
CS 224S W2006 33

$P(D)$

- Markov assumption
- Each dialogue act depends only on previous N . (In practice, N of 3 is enough).

■ Woszczyna and Waibel (1994):

$$P(D) = \prod_{i=2}^M P(d_i | d_{i-1}, \dots, d_{i-M+1})$$



CS 224S W2006 34

$P(W|D)$

- Each dialogue act has different words
- Questions have "are you...", "do you...", etc

$$P(W|D) = \prod_{i=2}^N P(w_i | w_{i-1}, \dots, w_{i-N+1}, d_i)$$

CS 224S W2006 35

$P(F|D)$

- 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)
 - Wh-Question (QW)

CS 224S W2006 36

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



Getting likelihoods from decision tree

- Decision trees give posterior $p(d|F)$ [discriminative, good]
- But we need $p(F|d)$ to fit into HMM

$$p(d|F) = \frac{P(F|d)p(d)}{p(F)}$$

- Rearranging terms to get a likelihood:

$$\nearrow \frac{p(F|d)}{p(F)} = \frac{P(d|F)}{p(d)}$$

- scaled likelihood is ok since $p(F)$ is constant

CS 224S W/2006

Final HMM equation for dialogue act tagging

$$D^* = \operatorname{argmax}_D P(D)P(F|D)P(W|D)$$

$$= \prod_{i=2}^M P(d_i | d_{i-1} \dots d_{i-M+1}) \prod_{i=1}^N \frac{P(d_i | F)}{P(d_i)} \prod_{i=2}^N P(w_i | w_{i-1} \dots w_{i-N+1}, d_i)$$

- Then can use Viterbi decoding to find D^*
- In real dialogue systems, obviously can't use FUTURE dialogue acts, so predict up to current act
- In rescoring passes (for example for labeling human-human dialogues for meeting summarization), can use future info.
- Most other supervised ML classifiers have been applied to DA tagging task

CS 224S W/2006

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.

CS 224S W/2006

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

CS 224S W/2006

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 <i>Um tomorrow</i>	NA
S You can say the departure date. For example, you can say 'today' or 'tomorrow'. Or you can say 'Monday', 'Tuesday', 'Wednesday'...	
U <i>Tomorrow</i>	REP
S You can say the departure date. For example, you can say 'today' or 'tomorrow'. Or you can say 'Monday', 'Tuesday', 'Wednesday'...	
U <i>Saturday</i>	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 o'clock p-m	
U <i>Eight o'clock</i>	OMIT
S Do you want to leave around ten o'clock p-m	
U <i>No</i>	OMIT
S What time do you want to leave?	
U <i>Eight o'clock A M</i>	ADD

Machine Learning and Classifiers

- Given a labeled training set
- We can build a classifier to label observations into classes
 - Decision Tree
 - Regression
 - SVM
- I won't introduce the algorithms here.
- But these are at the core of NLP/computational linguistics/Speech/Dialogue
- You can learn them in:
 - AI - CS 121/221
 - Machine Learning CS 229

CS 224S W2006 44

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)

CS 224S W2006 44

Generating Dialogue Acts

- Confirmation
- Rejection

CS 224S W2006 45

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

CS 224S W2006 46

Explicit confirmation

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

CS 224S W2006 47

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

CS 224S W2006 48

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?

CS 224S W2006

49

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).

CS 224S W2006

50

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

CS 224S W2006

51

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

CS 224S W2006

52

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 threshold, reject
 - Above threshold, explicit confirmation
 - If even higher, implicit confirmation
 - Even higher, no confirmation

CS 224S W2006

53

Dialogue System Evaluation

- **Key point about SLP.**
- Whenever we design a new algorithm or build a new application, need to evaluate it
- How to evaluate a dialogue system?
- What constitutes success or failure for a dialogue system?

CS 224S W2006

54

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

CS 224S W2006

55

Evaluating Dialogue Systems

- PARADISE framework (Walker et al '00)
- "Performance" of a dialogue system is affected both by *what* gets accomplished by the user and the dialogue agent and *how* it gets accomplished



Slide from Walker et al 2000

56

PARADISE evaluation again:

- Maximize Task Success
- Minimize Costs
 - Efficiency Measures
 - Quality Measures
- PARADISE (PARAdigm for Dialogue System Evaluation)

CS 224S W2006

57

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

CS 224S W2006

58

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 Jude Mitchell

CS 224S W2006

59

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

CS 224S W2006

60

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

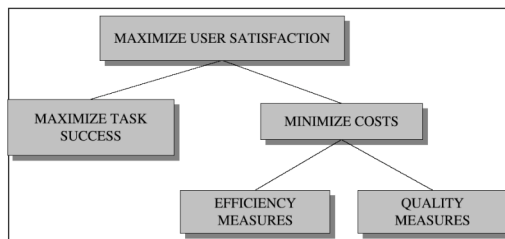
CS 224S W2006 61

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"

CS 224S W2006 62

PARADISE: Regress against user satisfaction



CS 224S W2006 63

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

CS 224S W2006 64

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

CS 224S W2006 65

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)

Adapted from Julia Hirschberg

Performance Functions from Three Systems

- $COMP_{user}(\pi) = \alpha \pi + COMP + (1 - \alpha) \cdot MRS + (1 - \alpha) \cdot ET$
- $MRS_{user}(\pi) = \beta \pi + COMP + (1 - \beta) \cdot MRS + \beta \cdot ET$
- $MRS_{system}(\pi) = \gamma \pi + COMP + (1 - \gamma) \cdot MRS + \gamma \cdot ET$

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 Dribinski

CS 224S W2006

67

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

CS 224S W2006

68

Now that we have a success metric

- Could we use it to help drive learning?
- We'll try to use this metric to help us learn an optimal policy or strategy for how the conversational agent should behave

CS 224S W2006

69

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* π for what action to take in any particular state.

CS 224S W2006

70

What do we mean by actions A and policies π ?

- 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?

CS 224S W2006

71

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

CS 224S W2006

72

Another strategy decision

- Open versus directive prompts
- When to do mixed initiative

CS 224S W2006 74

Review: 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"

CS 224S W2006 74

Review: Restrictive vs. Non-restrictive gramamrs

- 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

CS 224S W2006 75

Kinds of Initiative

- How do I decide which of these initiatives to use at each point in the dialogue?

Grammar	Open Prompt	Directive Prompt
Restrictive	<i>Doesn't make sense</i>	System Initiative
Non-restrictive	User Initiative	Mixed Initiative

CS 224S W2006 76

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 π* for what action to take in any particular state.

CS 224S W2006 77

Goals are not enough

- Goal: user satisfaction
- OK, that's all very well, but
 - Many things influence user satisfaction
 - We don't know user satisfaction til after the dialogue is done
 - How do we know, state by state and action by action, what the agent should do?
- We need a more helpful metric that can apply to each state
- We turn to Reinforcement Learning

CS 224S W2006 78

Utility

- A utility function
 - maps a state or state sequence
 - onto a real number
 - describing the goodness of that state
 - I.e. the resulting "happiness" of the agent
- Principle of Maximum Expected Utility:
 - A rational agent should choose an action that maximizes the agent's expected utility

CS 224S W2006

79

Maximum Expected Utility

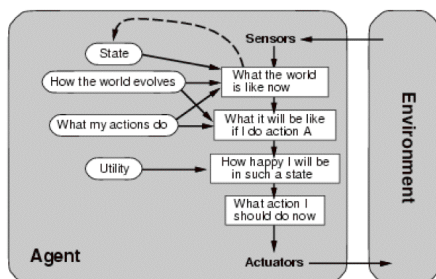
- Principle of Maximum Expected Utility:
 - A rational agent should choose an action that maximizes the agent's expected utility
- Action A has possible outcome states $Result_i(A)$
- E: agent's evidence about current state of world
- Before doing A, agent estimates prob of each outcome
 - $P(Result_i(A)|Do(A),E)$
- Thus can compute expected utility:

$$EU(A|E) = \sum_i P(Result_i(A)|Do(A),E)U(Result_i(A))$$

CS 224S W2006

80

Utility (Russell and Norvig)



CS 224S W2006

81

Markov Decision Processes

- Or MDP
- Characterized by:
 - a set of states S an agent can be in
 - a set of actions A the agent can take
 - A reward $r(a,s)$ that the agent receives for taking an action in a state
 - (+ Some other things I'll come back to (gamma, state transition probabilities))

CS 224S W2006

82

A brief tutorial example

- Levin et al (2000)
- A Day-and-Month dialogue system
- Goal: fill in a two-slot frame:
 - Month: November
 - Day: 12th
- Via the shortest possible interaction with user

CS 224S W2006

83

What is a state?

- In principle, MDP state could include any possible information about dialogue
 - Complete dialogue history so far
- Usually use a much more limited set
 - Values of slots in current frame
 - Most recent question asked to user
 - Users most recent answer
 - ASR confidence
 - etc

CS 224S W2006

84

State in the Day-and-Month example

- Values of the two slots day and month.
- Total:
 - 2 special initial state s_i and s_r .
 - 365 states with a day and month
 - 1 state for leap year
 - 12 states with a month but no day
 - 31 states with a day but no month
 - 411 total states

DS 2245 W2006

85

Actions in MDP models of dialogue

- Speech acts!
 - Ask a question
 - Explicit confirmation
 - Rejection
 - Give the user some database information
 - Tell the user their choices
- Do a database query

DS 2245 W2006

86

Actions in the Day-and-Month example

- a_d : a question asking for the day
- a_m : a question asking for the month
- a_{dm} : a question asking for the day+month
- a_r : a final action submitting the form and terminating the dialogue

DS 2245 W2006

87

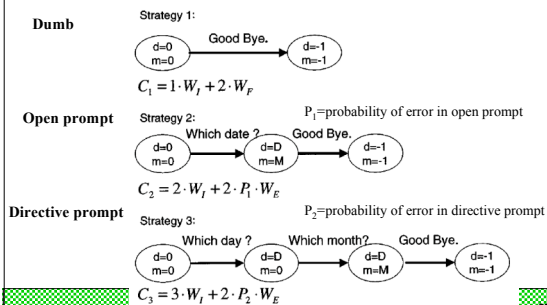
A simple reward function

- For this example, let's use a cost function
- A cost function for entire dialogue
- Let
 - N_i =number of interactions (duration of dialogue)
 - N_e =number of errors in the obtained values (0-2)
 - N_f =expected distance from goal
 - (0 for complete date, 1 if either data or month are missing, 2 if both missing)
- Then (weighted) cost is:
 - $C = w_i \times N_i + w_e \times N_e + w_f \times N_f$

DS 2245 W2006

88

3 possible policies



89

DS 2245 W2006

90

To be continued!

Summary

- Evaluation for dialogue systems
 - PARADISE
- Utility-based conversational agents
 - Policy/strategy for:
 - Confirmation
 - Rejection
 - Open/directive prompts
 - Initiative
 - +?????
 - MDP
 - POMDP

DS 2245 W2006

91

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

DS 2245 W2006

92