

# Evaluation of Spoken Dialogue Systems


## Three different dialogue systems

- Process-oriented
- Browsing
- Call-routing

# Functionality displayed

- Data elicitation
- Confirmation
- Recovery from error
  - Reprompt
  - Fallback to DTMF
- Flexibility in input
- Default action (possibly correct)?

## AMITIES, a Financial Service Application

- Hello, this is Amities. What can I do for you.
- Yes, this is Stella Lang. My card number is <CARD\_NUMBER>. Can you tell me the balance on my account.
- Can you just confirm the city and county?
- Yes, it's London, Middlesex.
- I'm sorry. I didn't understand what you said. Can you repeat please?
- London Middlesex.
- Alright, your balance is seventeen fifty three pounds and twenty pence. Anything else I can do for you?
- Yes, I'd like to make a payment of fifty three pounds with my debit card.
- Can I have the debit card number, Miss Lang?
- Sure, it's <CARD\_NUMBER> and it expires October, 2006. 
- What's the issue number please?
- One.
- Okay, Miss Lang. I'll send the payment amount of fifty three pounds. Did I get that right?
- Yes. What's my new balance please?
- Okay that's gone through. Your payment will be credited overnight and will reflect on your next statement. Your balance is seventeen hundred pounds and twenty pence. Anything else I can do for you today?
- No, that's it, thanks.
- Thanks very much for your call. Goodbye.
- Bye



## An Air-Travel Planning Application

- I'd like to return the following Tuesday.
- Okay, flights from Minneapolis St. Paul to Boston on Tuesday Nov. 27th. One moment please Can you provide a departure or arrival time? There are no non-stop flights. There are at least six connecting United flights from Minneapolis St. Paul to Boston on Tuesday Nov. 27<sup>th</sup>.
- I'd like to leave around 9:00 a.m.
- Okay, United flights from Min to Boston leaving at 9:00 a.m. on Tuesday, Nov. 27th. One moment please. What about a flight connecting in Chicago O'Hare leaving at 10:00a.m. Would that work? I also have a flight connecting in Chicago O'Hare at 8:00 a.m.
- 10:00 a.m.
- United flight 582 connected to United flight 510 in Chicago O'Hare has been added to your itinerary. You depart Minneapolis St. Paul at 10:00 a.m. and arrive in Boston at 2:56 p.m. Shall I price your itinerary?
- Pricing your itinerary will take a while. In the meantime, I'll summarize your itinerary for you. <ITINERARY>. The cheapest price for your itinerary is \$305.50. Shall I email your itinerary to you?
- Yes, please.
- Your itinerary has been emailed. To help us improve Mercury, please stay on the line for a moment to answer a few questions.

your itinerary. What date will you be returning on?

## Call Routing

- Goal of system: route call to the appropriate operator



## Question

- Given the very different systems you heard, what evaluation metrics do you think you could apply to help system developers
  - understand system behavior
  - assess user attitudes towards the system

## Goals of Evaluation

- Measure something that is correlated with success
- Require as little human input as possible
- Run continuously
- Provide feedback that is useful in making the system better

# Types of Metrics

- Objective, measuring
  - System performance, often at the module level
  - Success/task completion, measured by some objective goal
  - Efficiency
- Subjective, measuring
  - User perceptions

## Evaluation: Quantifying System Behavior

- Determining system accuracy
  - Have system components improved?
  - Does our system answer queries correctly?
  - Are subjects able to complete tasks?
- Verifying system integrity
  - Has a change in one system component adversely affected another?
  - Have we introduced a catastrophic bug?
- Understanding how our systems are used
  - Do our systems do what we claim they do?
  - Do our subjects use the system to solve real problems?
  - Are users satisfied with the results?

# Some Sample Evaluation Metrics

- Component evaluation:
  - Word/sentence evaluation
  - Parse coverage
  - Understanding evaluation
  - Paraphrase comparison
- System-level evaluation:
  - Static database assessment
  - Logfile evaluation

# Speech Recognition—Word Error Rate

- Defined as:
  - Insertions + Deletions + Substitutions
- Computed on word string
- Widely used in recognition community
- Most common metric for ASR performance

Hyp:

can you show flights from boston to denver on the afternoon of the sixteenth

Ref:

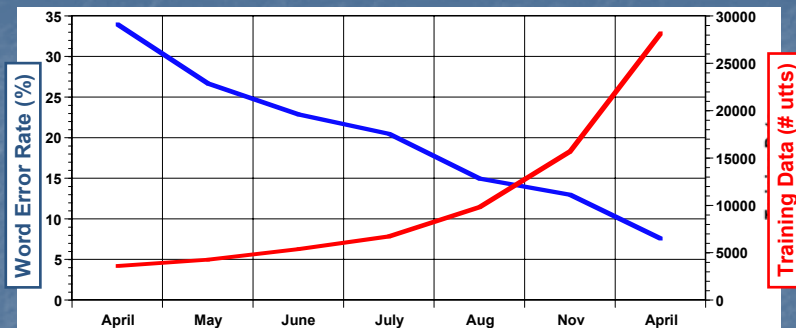
show me flights from boston to denver on the afternoon of june fifteenth

D D

I

S S

## Speech Recognition Evaluation-- Word Error as a Function of Data Collection



For In-Vocabulary Utterances only

- Increasing amounts of data have allowed for improved language and acoustic models
- Initial system trained on read speech
- Data collection from real users provides a better match between training and testing conditions

## Speech Recognition Evaluation: Sentence Error

- Sentence must be completely correct (no substitutions, insertions, or deletions)
- Usually much higher than word error and understanding error
- Can penalize irrelevant errors (e.g., "a" vs. "the", homophones)
- Sample performance on one weather domain test set: 34.5% (yet understanding rate is much higher)

# Understanding Evaluation: Concept Error

- Goal: automatic evaluation of understanding from meaning representation
- Motivation:
  - Component technologies interact, making evaluation of individual modules insufficient
  - Data coming in at a rapid pace, making examination of individual question/pairs impossible
  - System can offer information not specifically asked for, making evaluation of database tuples insufficient
- Mechanism: comparison of key\_value representation from semantic frame
- Computed in same way as Word Error:
  - Insertions + Deletions + Substitutions

# Understanding Evaluation: [key:value] Pairs

- Uses flat key-value representation for evaluation of understanding component
  - `WEATHER: rain CITY: boston DATE: tomorrow`
- Flattened structure evaluated for insertions, deletions, substitutions
  - Equivalence classes can be used to represent variants that do not affect understanding (e.g., 'weather: sun' and 'weather: cloudy' might map to a general statement about cloud cover)
  - Key-value pairs can be weighted by importance (e.g., city names)



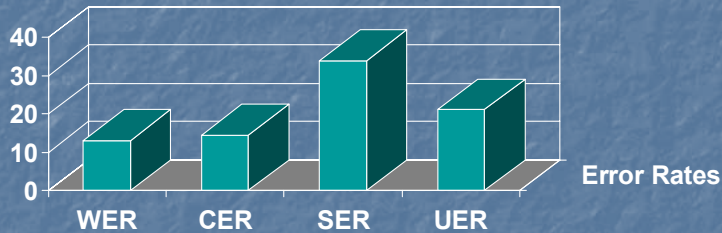
## How to evaluate the evaluation metric?

- Does it capture problematic utterances?
  - Can it identify places where attention is needed to fix problems?
- Is it correlated with an objective measure of performance (e.g., human rating)?

## Comparison with Human Evaluation

- Automated evaluation methodology compared against human evaluation
- Test set: 483 weather domain utterances
- 93% agreement on utterances judged "correct" by both methods
- Disagreements mainly on hard-to-evaluate cases:
  - Reference utterance unevaluable
  - Utterance correct because of pragmatics/discourse
  - Human judgement in error

## How metrics are related



- WER and CER, SER and UER correlated
- Difference between UER and SER can be accounted for partially by robust parsing

## Beyond Utterance-Based Evaluation Metrics: Dialogue-Based Evaluation

- Utterance-based metrics do not measure understanding in the context of a dialogue
- Dialogue-based metrics can potentially quantify:
  - How effectively a user can convey concepts to the system
  - How efficiently a system understands concepts
- A simple way of viewing dialogues and their associated parameters can be very useful
- Automatic procedures necessary to keep up with large amounts of data

## Issues in Evaluating Dialogue Data: Consistency

- If *answer* is the key, what happens when answer changes?
  - What's the cheapest flight from London to Boston?
  - Will it rain tomorrow in Sheffield?
  - Is there a good Chinese restaurant in Walkley?

Dynamic data result in same query having different answers on different days

## Issues in Evaluating Dialogue Data: Changes in System Capabilities

Dialogue 1:

User: I want to fly from Boston to Alexandria, Louisiana

System: I'm sorry, I didn't understand what you said.  
Please repeat your query.

User: I want to fly from Boston to Alexandria, Louisiana

System: I'm sorry, I didn't understand what you said.  
Please repeat your query.

User: Boston to Alexandria. Flights.

System: I'm sorry, I didn't understand what you said.  
Please repeat your query.

## Issues in Evaluating Dialogue Data: Changes in System Capabilities (simple change)

### Dialogue 2:

User: I want to fly from Boston to Alexandria, Louisiana.

System: Okay, flights from Boston to Alexandria. On what date would you like to leave?

User: I want to fly from Boston to Alexandria, Louisiana.

System: Okay, flights from Boston to Alexandria. On what date would you like to leave?

User: Boston to Alexandria. Flights.

System: Okay, flights from Boston to Alexandria. On what date would you like to leave?

## Issues in Evaluating Dialogue Data: Changes in System Capabilities (more complex change)

### Dialogue 3:

User: I want to fly from Boston to Alexandria, Louisiana.

System: I think you want to fly to a city in Louisiana. Please speak and spell the name of the city you wish to fly to.

User: I want to fly from Boston to Alexandria, Louisiana.

System: I think you want to fly to a city in Louisiana. Please speak and spell the name of the city you wish to fly to.

User: Boston to Alexandria. Flights.

System: I'm sorry I didn't understand you. Please repeat your query.

# Issues in Evaluating Dialogue Data: Data

- How do we use old data to test new systems/functionality (without spending so much time on the data themselves that we don't have time to evaluate)?
- How do we make use of large amounts of data?
  - Freezing databases
    - Inconvenient for large databases
    - Doesn't test new functionality
  - Adding contrived queries to get incoherent dialogues back on track/test new functionality
    - Very complicated
- What best represents the correct answer?
  - Key-value representation useful for complex dialogues
    - Each incoming query evaluated against the "truth"
  - Answer frames or strings (requires frozen database)

## Dialogue-Based Evaluation

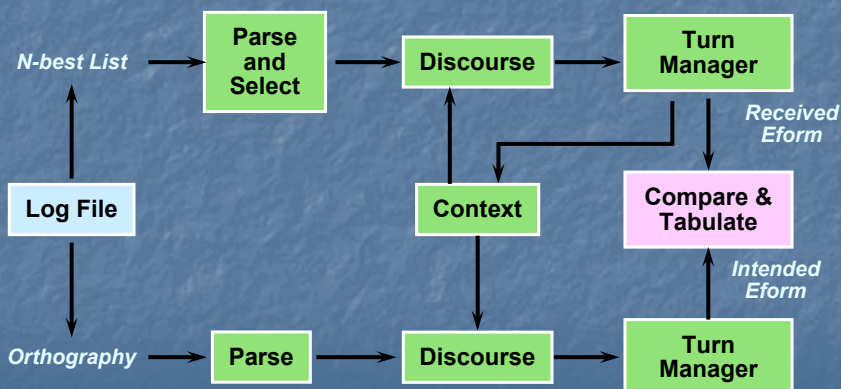
- **Concept Efficiency** [frustration measure]
  - On average, how many times did the user repeat a concept before the system understood it?
- **Query Density** [information transfer measure]
  - On average, how many concepts does the user communicate to the system per utterance spoken?
- Concepts monitored *after* turn management is completed
  - Assesses recognition, understanding, discourse, and dialogue
- Both measures require a re-processing of log files after orthographic transcription has been supplied
- Measures automatically computed
- Requires specification of concepts for each domain

# What you need to keep track of/know

- List of concepts to be monitored (e.g., source, destination, date, time)
- What the user actually said (i.e., an orthographic transcription of utterance)
- What the system did
  - If reusing data, user queries are based on what happened at time of data collection
  - To understand how long it took to get a concept across NOW vs. THEN, you have to know how long it took THEN and how long it takes NOW
  - To get an idea of system performance NOW vs. THEN, you need to keep both contexts around

# Regression Testing Dialogue Data

- In practice, a bit tricky, but it can be automated



# Concept Efficiency

**U:** I want to go to Boston  
**S:** TO BOSTON FROM WHERE

**U:** I want to go to Boston  
**S:** TO AUSTIN FROM WHERE  
**U:** No, Boston, Massachusetts  
**S:** TO BOSTON FROM WHERE

- It took one user utterance to convey one concept in the left example, and two user utterances in the right example

$$CE = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{N_u(i)}{N_c(i)}$$

$N_d$  = Number of dialogues

$N_c(i)$  = total number of concepts in the  $i^{th}$  dialogue

$N_u(i)$  = number of *unique* concepts understood by system in the  $i^{th}$  dialogue

# Query Density

**S:** WHERE WOULD YOU LIKE TO GO  
**U:** I want to go from Boston to Denver  
**S:** ON WHAT DATE

**S:** WHERE WOULD YOU LIKE TO GO  
**U:** Denver  
**S:** WHERE ARE YOU LEAVING FROM  
**U:** Boston  
**S:** ON WHAT DATE

- Two concepts were conveyed in one user utterance in the left example, and one in the right example

$$QD = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{N_u(i)}{N_q(i)}$$

$N_d$  = Number of dialogues

$N_q(i)$  = total number of user queries in the  $i^{th}$  dialogue

$N_u(i)$  = number of *unique* concepts understood by system in the  $i^{th}$  dialogue

# CE and QD in Practice

- CE on a subset of flight domain utterances:  
**0.91**
  - Approximately 1 in 10 concepts had to be repeated
  - High concept efficiency correlated with high user satisfaction?
- QD on subset of flight domain utterances:  
**1.7**
  - Subjects are entering more than one concept per turn
  - Lower query density correlated with degree of system initiative?

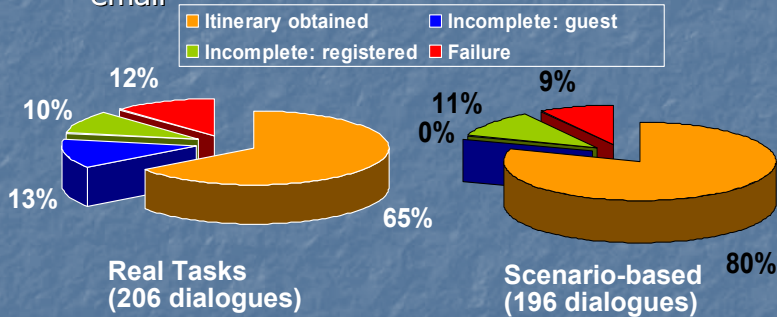
# System Diagnostics from Logfiles

- Metrics must be automatic and provide easy-to-access, daily summaries to system developers
- Dialogue-level diagnostics
  - Task completion
  - Characteristics of user/system behavior (e.g., repeated responses, user difficulty in conveying particular concepts)
- Utterance-level diagnostics
  - User/system confusion (e.g., keypad entries)
  - Number of help requests
- Computed off-line from logfile



## Measuring Task Performance from Logfiles (Air Travel Domain)

- Computed completely automatically
  - Logfiles parsed for self-consistent legs, pricing, and email



- Incomplete: self-consistent legs but incomplete itinerary
  - Guest users often just shop for possible flight times
  - Registered user dialogues in this category revealed the most interesting problems

## Conclusion

- Evaluation cannot be ignored in SDS development
- Suite of evaluation metrics should be in place before system is deployed
- Once system is running, evaluation metrics must be automatic and ongoing
- User satisfaction is the final answer, but it's the most difficult to collect