Corrections & Repairs

ErikAnthony Harté
Ling 575 - Spring 2013

SDS Errors in Understanding

Mismatch between action intended vs. action taken

- ASR (automatic speech recognition)
- NLU (natural language understanding)

Prevent

- improve ASR
- simplify tasks
- constrain domain or vocabulary

Detect/Correct

- ???

Why Bother?

An efficient error handling strategy could allow our SDS to

- change system initiative strategy
- change dialog strategy
- modify ASR function

For example, upon detection of a possible misunderstanding the system might switch to an ASR function better tuned to recognize hyperarticulated speech.

Outline

1. Detection
   Can prosodic features be used to recognize corrections?
2. Correction
   A quick look at the RavenClaw Architecture for error handling.
3. A Sample Strategy
   Using dialog costs to determine the optimum grounding strategy.

Detection: Experiment

TOOT - Phone based, train information dialog system

- 2528 turns, 152 dialogs
- Initiative:
    - system, user, mixed
- Confirmation:
    - implicit, explicit
- Strategy:
    - adaptive, non-adaptive
- Concept Accuracy - ASR task information recognition
- Word Error Rate

Detection: Types of Corrections

REP - Repetitions	39%
OMT - Omissions	31%
PAR - Paraphrasing	19%
ADD - Additions	8%
A/O - Additions & Omissions	2%

Detection: Hyperarticulation associated with corrections?

Hyperarticulation:

- slower
- louder
- higher pitch
- follows longer pauses
- greater internal silence

Features

- f0 - fundamental frequency
- RMAX - energy
- duration
- length of preceding pause
- speaking rate

Result: 58% of corrections vs 12% non-corrections

Detection: Machine Learning

Feature set selected for generating classifier:

Prosodic (PROS) :

Raw (raw values): f0max, f0mn, rmsmax, rmsmn, dur, ppau, tempo, zeros
Norm1 (values normalized by first turn in dialogue): f0max1, f0mn1, rmsmax1, rmsmn1, dur1, ppau1, tempo1, zeros1
Norm2 (values normalized by previous turn in dialogue): f0max2, f0mn2, rmsmax2, rmsmn2, dur2, ppau2, tempo2, zeros2

ASR (ASR) : gram, str, conf, ynstr, nofeat, canc, help, wordsstr, syls, rejbool

System Experimental (SYS) : inittype, conftype, adapt, realstrat

Dialogue Position (POS) diadist

Dialogue History (DIA) :

PreTurn : value of PROS and ASR features for preceding turn (e.g., pref0max)
PrepreTurn : value of PROS and ASR features for turn preceding preceding turn (e.g., ppref0max)
Prior : for each Boolean-valued feature (ynstr, nofeat, canc, help, rejbool), the number/percentage of prior turns exhibiting the feature (e.g., priorynstr-num/priorynstrpct)
PMean : for each continuous-valued PROS and ASR feature, the mean of the feature's value over all prior turns (e.g., pmnf0max)

Figure 3
Feature set for predicting corrections.

Detection: Machine Learning Results

The best feature set saw a reduction in the error rate from 29% to 15.72%

Table 7
Estimated error, recall, precision, and $F_\beta = 1$ for predicting corrections.

			class = T			class = F		
Features	DIA	Error ± SE	Rec.	Prec.	$F_\beta = 1$	Rec.	Prec.	$F_\beta = 1$
Raw+ASR+SYS+POS	PreTurn	15.72 ± 0.80	70.61	74.96	.72	89.95	88.28	.89
Raw+ASR+SYS+POS	all	16.16 ± 0.58	69.80	74.65	.72	90.12	87.82	.89
PROS+ASR+SYS+POS	all	16.38 ± 0.61	69.01	74.05	.71	89.60	87.61	.88
ASR	all	16.41 ± 0.93	69.93	72.39	.70	88.76	87.7	.88
ASR+SYS+POS	all	17.01 ± 0.78	73.73	73.38	.73	88.68	89.00	.89
ASR+SYS+POS	none	18.60 ± 0.81	56.48	72.79	.63	91.33	83.76	.87
Raw+ASR+SYS+POS	none	18.68 ± 0.67	58.45	71.64	.64	90.37	84.17	.87
ASR+PROS	none	19.29 ± 0.78	54.54	69.97	.61	90.25	82.90	.86
POS+PROS	none	19.59 ± 0.73	52.96	69.70	.60	90.38	82.47	.86
Raw	all	19.68 ± 0.78	55.62	70.89	.62	90.64	83.33	.87
PROS	all	20.33 ± 0.90	56.45	69.23	.61	89.43	83.42	.86
ASR+POS	none	20.40 ± 0.79	52.20	71.99	.60	91.43	82.41	.87
PROS	none	20.53 ± 0.81	54.86	71.72	.62	90.78	83.07	.87
conf+rejbool	all	21.23 ± 0.93	59.70	65.97	.62	87.05	84.05	.85
ASR+SYS	none	23.46 ± 0.72	51.55	63.40	.56	87.53	81.65	.84
ASR	none	24.19 ± 0.84	45.93	60.99	.52	87.80	79.90	.84
Raw	none	25.35 ± 0.93	42.26	59.46	.48	88.29	78.97	.83
POS	none	29.00 ± 1.02	0.00	–	–	99.94	70.99	.83
SYS	none	29.00 ± 1.02	0.00	–	–	100.00	71.00	.83

Prerejbool baseline error = 25.70; majority baseline error = 28.99

Correction: RavenClaw Framework

Requirements for Detection/Correction

1. ability to detect errors
2. set of recovery strategies
3. mechanism for selection and employing strategies

Domain Specific

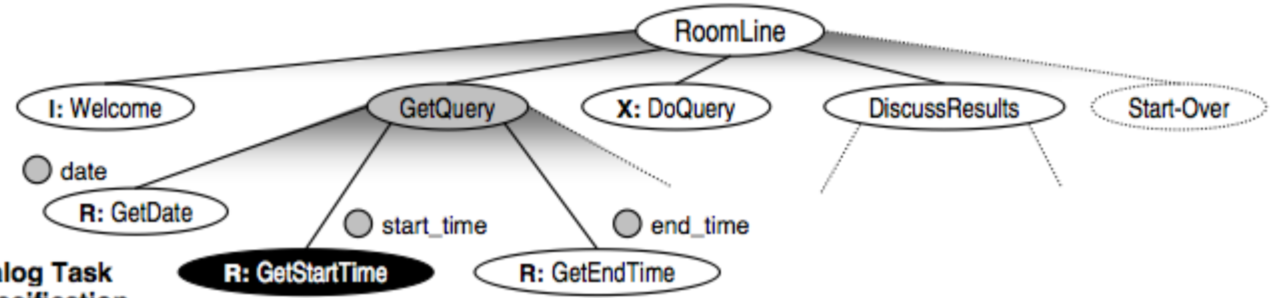
- Dialog Task Specification

Domain Independent

- Dialog Engine
  - error handling
  - timing
  - turn tracking

Correction: Dialog Task Specification

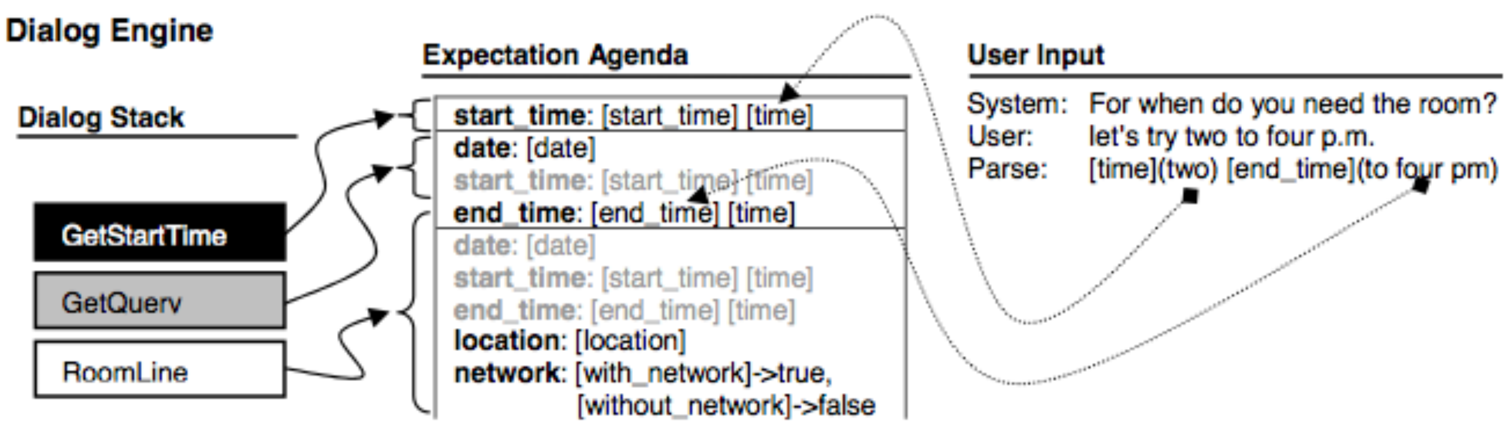
- Each *agent* manages a subpart of the dialog.
- Information is captured in *concepts*.
- Each leaf-level agent is associated with a specific concept.
- Four types of agent: *Inform*, *Request*, *Expect*, *Execute*



Dialog Task Specification

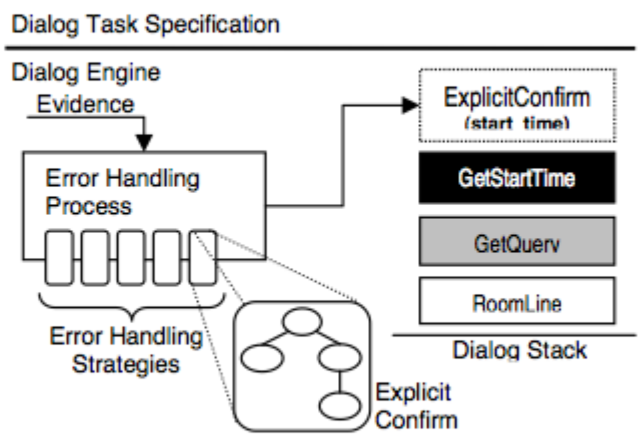
Correction: Dialog Engine

- Manages Dialog
- Dialog Stack + Expectation Agenda



Correction: Error Handling

- Error Handling (EH) process has a set of strategies
- Each concept and each basic agent in the DTS gets its own EH process
- All EH processes run simultaneously: A gating processes determines which process gets placed on top of stack



Correction: Error Handling Strategies

Misunderstanding vs. Non-understanding

- Incorrect sematic interpretation => leads to action but not likely correct
- No interpretation => no action, but still negative impact on quality of interaction

Misunderstanding Strategies

- *explicit confirm,implicit confirm,reject*

Non-understanding Strategies

- *ask repeat, ask rephrase, reprompt, detailed reprompt, notify, yield, moveOn, youCanSay, fullHelp*

A Sample Strategy

Grounding

The exchange of positive and negative evidence to reduce uncertainty in the dialog

Kinds of evidence

- Display (implicit)
- Clarify (explicit)

...also

- Reject
- Accept

(1) U: I can see a red building.
S (ACCEPT): *Ok, can you see a tree in front of you?*
S (DISPLAY): *Ok, a red building, can you see a tree in front of you?*
S (CLARIFY): *A red building?*
S (REJECT): *What did you say?*

A Sample Strategy

How to decide what kind of evidence to provide?

- Level of uncertainty
- Task related costs and utility
- Cost of grounding action

Typical: Examine ASR confidence score

- High - *Accept*
- Mid - *Display*
- MidLow - *Clarify*
- Low - *Reject*

But, this only looks at one of the three factors...

A Sample Strategy

Principle of Maximal Expected Utility, MEU

- *Choose a grounding action (GA), so that the sum of all task-related costs and grounding costs are minimized considering the probability that the recognition hypothesis is correct.*

$$GA = \text{argMin}(a)\ \{\ P(\text{correct}) * \text{Cost}(a,\text{correct}) + P(\text{incorrect}) * \text{Cost}(a,\text{incorrect})\ \}$$

A Sample Strategy

- For *P(correct)* we can use the ASR conf score...But,

  ...We still need *Cost(a, correct)*, and *Cost(a,incorrect)*
- Ultimate measure of *Cost(a,incorrect)* is the reduction in user satisfaction, but that is at dialog level, we need turn level.
- **Efficiency**. "All things being equal agents try to minimize their effort at inducing what what intend to do."

Total number of syllables uttered.

A Sample Strategy

Grounding Action Costs

- Example: Cost for choosing ACCEPT incorrectly: *Number of extra syllables needed to later correct the dialog.*

Table 1: Costs for different grounding actions, given the correctness of the recognition (COR=Correct, INC=Incorrect).

Action,Hyp	**Costs**
ACCEPT,COR	No cost
ACCEPT,INC	The number of extra syllables the misunderstanding adds to the dialogue (*SylMis*).
DISPLAY,COR	Grounding dialogue *(SylDispCor)*.
DISPLAY,INC	Grounding dialogue (*SylDispInc*). Risk that the user does not correct the system *(P(Fail|Disp,Inc))* times the consequences of a misunderstanding (*SylMis*).
CLARIFY,COR	Grounding dialogue *(SylClarCor)*. Risk that the user does not confirm the system *(P(Fail|Clar,Cor))* times the syllables for recovering the rejected concept *(SylRec)*.
CLARIFY,INC	Grounding dialogue *(SylClarInc)*
REJECT,COR	The number of syllables it takes to receive new information of the same value as the rejected concept *(SylRec)*.
REJECT,INC	No cost

A Sample Strategy

Example: A short correction dialog - two syllables

S: *Red?*

U: Yes.

Table 2: Cost functions for different grounding actions.

Action	Expected cost
ACCEPT	*P(incorrect)* x *SylMis*
DISPLAY	*P(correct)* x *SylDispCor* + *P(incorrect)* x (*SylDispInc* + *P(Fail|Disp,Inc)* x *SylMis*)
CLARIFY	*P(correct)* x (*SylClarCor* + *P(Fail|Clar,Cor)* x *SylRec*) + *P(incorrect)* x *SylClarInc*
REJECT	*P(correct)* x *SylRec*

Questions

?

References

1) Diane Litman; Marc Swerts; Julia Hirschberg. (2006)
Characterizing and Predicting Corrections in Spoken Dialogue Systems
Computational Linguistics, 32(3):417-438.

2) Dan Bohus; Alexander Rudnicky. (2005)
Error Handling in the RavenClaw Dialog Management Architecture
In Proceedings of HLT-EMNLP 2005, p. 225-232

3) G. Skantze. (2007)
Making grounding decisions:Data-driven estimation of dialogue costs and confidence thresholds
In Proceedings of the 8th SIGdial workshop on Discourse and Dialog, p.206-210

Thank You!