Empirically Evaluating an Adaptable Spoken Dialogue System

Diane J. Litman

AT&T Labs - Research
Florham Park, NJ USA
http://www.research.att.com/info/diane

Shimei Pan

Columbia University
New York, NY USA
http://www.cs.columbia.edu/~pan

Overview

- Problem
  - real-time spoken dialogue systems have many limitations, e.g., vocabulary, grammar, and application functionality
  - when users do not respect such limitations, performance typically degrades
- Hypothesis
  - performance can be improved by better matching a system's dialogue behaviors to a user's understanding of system limitations
- Results
  - performance is indeed improved when users are allowed to dynamically adapt a dialogue system's behavior

Dialogue with User Initiative, No Confirmation

Toot:	Hi, this is AT&T Amtrak schedule system. This is TOOT. How may I help you?
User:	I want the trains from New York City to Washington DC on Monday at 9:30 in the evening.
Toot:	Do you want me to find the trains from New York City to Washington DC on Monday approximately at 9:30 in the evening now?
User:	Yes.
Toot:	I am going to get the train schedule for you ...

Another User Initiative, No Confirmation

U1: I'd like to go to Chicago from Baltimore on Saturday at <u>8 am</u>.

T1: Do you want to leave <u>in the morning or in the evening</u>?

U2: `Uh morning`.

T2: **`Sorry, I can't understand you.`** Please repeat your answer.

U3: In the morning.

T3: Do you want me to find the trains from Baltimore to Chicago on Saturday around <u>10</u> morning now?

U4: No.

T4: How may I help you?

U5: I would like to go from Baltimore to **Chicago on Sat. at 8 o'clock**.

T5: Do you want me to find the trains from Baltimore to **DC on Tuesday around 11:45 am now?**

U6: No.

T6: How may I help you?

U7: *Change strategy.*

T7: *You are using the user initiative. Which initiative do you want to change to? System –*

U8: *System initiative please.*

T8: *You are using the no confirmation strategy. Which confirmation strategy do you want to change to? No confirmation, implicit confirmation, or explicit confirmation?*

U9: *Implicit please.*

T9: I heard you say go to DC on Tuesday around 11:45 am. Do you want me to find the trains from Baltimore to DC on Tuesday around 11:45 am now?

U10: No.

T10: Which city do you want to go to?

U11: Chicago.

T11: I heard you say go to Chicago. Which city do you want to leave from?

Dialogue w/ System Initiative, Explicit Confirmation

Toot:	Welcome to AT&T Amtrak schedule system. This is TOOT. Which city do you want to go to?
User:	Washington DC.
Toot:	Do you want to go to Washington DC?
User:	*Change strategy.*
	...
	(user changes to user initiative, implicit confirmation)
	...
Toot:	I heard you say go to Washington DC. Which city do you want to leave from?
User:	From New York City on Monday at 9:30 pm.
Toot:	I heard you say leave from New York City on Monday approximately at 9:30 pm. Do you want me to find the trains from New York City to Washington DC on -
User:	Yes.
Toot:	I am going to get the train schedule for you ...

An Evaluation of Adaptability in TOOT: Experimental Design

- 4 versions of TOOT
  - adaptable (by user) and non-adaptable
  - “System Initiative with Explicit Confirmation” and “User Initiative with No Confirmation”
- 5 subjects (naive users) per TOOT
- 4 tasks (similar difficulty) per subject
- 80 total dialogues (recordings, logs, and user surveys)
  - 2633 turns
  - 5.4 hours of speech
- implemented using internal spoken dialogue platform
  - automatic speech recognition (ASR)
  - text-to-speech (TTS)
  - phone interface
  - dialogue manager
  - application manager

Sample Task

- **Scenario Description**
  - Try to find a train going to **Philadelphia** from **New York City** on the **weekend** at **4:00 pm.** If you cannot find an exact match, find the one with the **closest** departure time. Please write down the **exact departure time** of the train you found as well as the **total travel time.** ("weekend" means the train departure date includes either Saturday or Sunday)
- **Hints**
  - If you don't know what to say or what the system is doing, say **Help** to hear a help message.
  - ...

Dialogue Efficiency Measures (automatically logged)

- User Turns
- System Turns
- Elapsed Time

Dialogue Quality Measures
(automatically logged, except where noted)

- Speech Recognition Rejections
- Timeouts
- Help Requests
- Barge Ins (manually labeled)
- ASR Rate (manually labeled)

Task Success Measure
(computed from logs and user surveys)

- Please enter the answers you found below.
  - Exact Departure Time:
  - Total Travel Time:

User Satisfaction: Sum of Many Measures (computed from user surveys)

- **TTS Performance**
  - Was the system easy to understand in this conversation?
    - VeryEasy SomewhatEasy NeitherEasyNorDifficult SomewhatDifficult VeryDifficult
- **Speech Recognition Performance**
- **Task Ease**
- **Interaction Pace**
- **User Expertise**
- **System Response**
- **Expected Behavior**
- **Future Use**

Results: Methodologies

- *Analysis of variance* [ANOVA] to determine whether adaptability produces significant differences among evaluation measures
  - mixed experimental design
  - 3 factors (adaptability, initial dialogue strategy, and task)
- *PARADISE* [Walker, Litman, Kamm, and Abella 1997] to estimate a performance function from evaluation measures
  - performance as subjective user satisfaction
  - model performance from remaining objective measures using multiple linear regression

Adaptability Effects

- Main effects of adaptability ($p<.03$)

Measure	Non-Adaptable (n=40)	Adaptable (n=40)
Task Success (%)	55.0	*80.0*
User Satisfaction	26.7	*31.6*

- Interaction effects of adaptability and initial dialogue strategy ($p<.05$)

	Non-Adaptable (n=40)		Adaptable (n=40)	
Measure	SystemExplicit	UserNo	SystemExplicit	UserNo
ASR Rate (%)	*88.4*	57.9	82.5	*75.8*
User Expertise	*4.7*	3.0	4.4	*3.8*
Future Use	3.5	1.7	*3.6*	*3.8*

- Summary
  - adaptable TOOT generally outperforms non-adaptable TOOT
  - the utility of adaptation depends on initial dialogue strategy

Contributors to Performance

- Performance function derived using PARADISE evaluation framework (accounts for 55% of the variance)

$$UserSat. = .45\mathcal{N}(ASR) + .33\mathcal{N}(Success) - .14\mathcal{N}(Time)$$

- Summary
  - predicts that more accurate speech recognition, more task success, and shorter dialogues will increase user satisfaction
  - helps explain the main effect of adaptability for User Sat.
  - draws into question assumptions about the centrality of efficiency

Related Work

- *Utterance* level adaptation
  - initiative adaptation [van Zanten 98]
  - confirmation adaptation [Smith 98]
- *Dialogue* level adaptation
  - automate adaptation by recognizing poor ASR performance [Litman, Walker, and Kearns 99]
- Adaptation over *multiple* dialogues
  - optimal strategies via reinforcement learning [Levin and Pieraccini 97], [Walker et al. 98], [Satinder, Kearns, Litman, and Walker, submitted]
- *User* rather than system adaptation
  - novices exhibit expert behavior after short tutorials [Kamm, Litman, and Walker 98]
- Adaptation in *software systems*
  - user modeling improves user satisfaction [Strachen, Anderson, Sneesby, and Evans 97]

Current Directions

- Generalization of results to other initial dialogue strategies
- Evaluation of *adaptive* (rather than adaptable) TOOT
  - user modeling based on ASR confidence
  - when and how to adapt learned from current data

Summary

- Builders of spoken dialogue systems face fundamental design choices that strongly influence system performance
- Users can improve system performance by *adapting* the system to the circumstances at hand
  - a variety of differences depend on adaptability
  - the utility of adaptability varies with dialogue strategy
  - the best performance predictors show adaptability effects

Detecting Poor ASR at the Dialogue Level: A Machine Learning Approach

(joint work with Marilyn Walker and Michael Kearns)

- Automate the adaptation process, by detecting and reacting to poor speech recognition
  - poor speech recognition is most predictive of performance
  - classifiers for poor performance can be learned from corpora
  - recognition performance can be improved by changing dialogue strategies
- 2 Classifications
  - if (% of misrecognized utterances > threshold) then (dialogue = "bad"); else (dialogue = "good")
  - objective definition, but requires hand-labeling
- 23 Features
  - acoustic, efficiency, quality, lexical, experimental
  - automatically computable from system logs

Efficiency + Quality + Experimental Vars Rules (73.4% accuracy vs. 52% baseline)

---

if (cancel% $\geq$ 4) $\wedge$ (system = toot) **then** *bad*
if (system turns $\geq$ 26) $\wedge$ (rejection% $\geq$ 5) **then** *bad*
if (condition = mixed) $\wedge$ (user turns $\geq$ 12) **then** *bad*
if (system = toot) $\wedge$ (user turns $\geq$ 14) **then** *bad*
if (cancels $\geq$ 1) $\wedge$ (timeout% $\geq$ 11) **then** *bad*
if (elapsed time $\leq$ 87 secs) **then** *bad*
default is *good*