Issue Based Dialogue Management, part 3

Staffan Larsson
Dialogue Systems II
Master in Language Technology programme
University of Gothenburg

September 27, 2018

Outline

Grounding in IBiS/TDM

- Grounding according to Clark and Schaefer (1989): the process of adding to the common ground
- More generally, the process of coordinating common ground, i.e. making sure that the participants' "take" on the common ground are sufficiently similar to enable communication
- In IBiS, grounding involves the following components:
  - Dialogue moves regulating coordination of the dialogue state (the SHARED field)
  - Information state update rules for selecting and integrating grounding-related dialogue moves
  - Update rules for integrating the content of core dialogue moves (ask, answer, request) into the dialogue state

Interactive Communication Management

- Communication management
  - Own Communication Management (OCM): self-corrections, hesitations, etc.
  - Interactive Communication Management (ICM): communication dealing with the management of dialogue interaction
- We will use the term ICM as a general term for coordination of the common ground, i.e. explicit signals (e.g., utterances) enabling coordination of updates to the common ground.

ICM categories

- Feedback
  - In general, behaviour whose primary function is to deal with grounding of specific utterances in dialogue
  - Utterances which signal grounding status of previous utterance
  - “mm”, “right”, “ok”, “pardon?”, “huh?” etc.
- Sequencing
  - Utterances which signal dialogue structure, and reflecting updates to the dialogue gameboard
  - “so”, ”now”, ”right”, ”anyway” etc.
- Turntaking

ICM in commercial systems

- Usually, limited to "verification"
- Examples (San Segundo et. al. 2001)
  - I understood you want to depart from Madrid. Is that correct? ["explicit verification"]
  - You leave from Madrid. Where are you arriving at? ["implicit verification"]
- Involves repetition or reformulation
- Appears in H-H dialogue, but not very common

From verification to ICM in dialogue systems

- "Verification" is just one type of ICM behaviour
  - Perhaps the one most crucial in dialogue systems given poor speech recognition
- Could a wider range of the ICM behaviour occurring in H-H dialogue be useful in dialogue systems?
- We want a typology of ICM moves for H-H dialogue
  - Here, we focus on feedback and sequencing moves
- We want to formalise it and use it in a system
  - Still we will implement only a subset
- We want to relate it to grounding in a system

Outline

Outline

Classifying feedback

- Level of action
- Polarity
- Eliciting or non-eliciting
- Form (syntactic realisation)
- Content type (object- or metalevel)

Feedback levels

Action levels in dialogue (Allwood, Clark, Ginzburg)

- Contact: whether a channel of communication is established
- Perception: whether DPs are perceiving each other's utterances
- Understanding: Whether DPs are understanding each other's utterances
  - Non-contextual ("semantic") meaning
  - Contextual ("pragmatic") meaning
- Acceptance: Whether DPs are accepting each other's utterances
- The function of feedback is to signal the status of utterance processing on all levels

Feedback polarity

- Polarity
  - Positive: indicates contact, perception, understanding, acceptance
  - Negative: indicates lack of contact, perception, understanding, acceptance
- We add a "neutral" or "checking" polarity - there is one or more hypotheses, but the DP lacks confidence in them
- Examples
  - "I don't understand": negative
  - "To Paris, is that correct?": checking
  - "To Paris.": positive
  - "Pardon": negative

Eliciting / noneliciting feedback

- Eliciting feedback is intended to evoke a response from the user
- Noneliciting feedback is not so intended, but may nevertheless receive a response

Object- or metalevel content

- Utterances with metalevel content explicitly refer to contact, perception, understanding or acceptance
- Object-level utterances instead refer to the task at hand
- Example
  - S: What city are you going to?
  - U: Paris
  - S(1a): Did you say you're going to Paris? [meta]
  - S(1b): Are you going to Paris? [object]
  - S(2a): Do you mean Paris, France or Paris, Texas?
  - S(2b): Do you want to go to Paris, France or Paris, Texas?
- This dimension does not apply to all feedback, e.g. "Paris.", "Pardon?"
  - (Is 2b feedback or simply an alternative question?)

Sentence type

- Syntactic form:
  - declarative: "I didn't hear what you said."; "The destination city is Paris."
  - interrogative: "What did you say?"; "Do you want to go to Paris?"
  - imperative: "Please repeat your latest utterance!"
  - elliptical
    - interrogative: "Paris?", "To Paris or from Paris?"
    - declarative: "To Paris."
- In general, the exact formulation of ICM phrases may depend on various contextual factors including activity, noise level, time constraints etc.

Simplifying assumptions regarding feedback

- We only represent action level and polarity
- Syntactic form not included; decided by the grammar
- Eliciting/noneliciting dimension replaced by the notion of "question-raising"- whether the utterance poses a question that should be addressed
  - Checking feedback is question-raising, since it poses a question that must be addressed ("Did you say/mean X?"; "X, is that correct?")
  - Positive feedback is not question-raising (although it may be responded to; more on this later)
  - Negative feedback (e.g. "I didn't hear what you said") is currently not question-raising, although arguably sometimes it should be ("What did you say?")

Formalising ICM dialogue moves

Level

- con: contact
- per: perception
- sem: semantic understanding (no context)
- und: pragmatic understanding (relevance in context)
- acc: acceptance

Polarity

- pos: positive
- neg: negative
- chk: checking

Feedback move notation

- icm:*Level* $*$ *Polarity*$\{$:*Args*$\}$
- Examples
  - icm:per$*$pos:*String* - "I heard 'to Paris' "
  - icm:und$*$neg - "Sorry, I don't understand"
  - icm:und$*$pos:*P* - "To Paris."
  - icm:acc$*$neg:*Q* - "Sorry, I can't answer Q"
  - icm:acc$*$pos - "Okay"

System feedback for user utterances in GoDiS

- contact
  - negative (icm:con∗neg): "I didn't hear anything from you.", "Hello?"
- perception
  - negative (icm:per∗neg): "Pardon?", "I didn't hear what you said"
  - positive (icm:per∗pos:*String*): "I heard 'to Paris'."
- semantic understanding:
  - negative (icm:sem∗neg): "I don't understand"
  - positive (icm:sem∗pos:*Content*): "Paris."

- pragmatic understanding
  - negative (icm:und∗neg): "I don't quite understand"
  - positive (icm:und∗pos:*Content*): "To Paris."
  - checking (icm:und∗chk:*Content*) "To Paris, is that correct?", "To Paris?"
- acceptance/integration
  - negative (icm:acc∗neg:*Content*): "Sorry, I cannot answer Q", "Sorry, Paris is not a valid destination city."
  - positive (icm:acc∗pos): "okay."

User feedback for system utterances in GoDiS

- contact: -
- perception
  - negative (icm:per∗neg): "Pardon?", "I didn't hear what you said"
- understanding: -
- acceptance/integration
  - negative (icm:acc∗neg): "I don't know", "Never mind"
  - positive (icm:acc∗pos): "okay."

Outline

Selecting feedback: Thresholds

- Four threshold levels are defined to enable selecting feedback
  - High: system interpretation is highly likely to be correct
  - Medium: system interpretation is moderately likely to be correct
  - Low: system interpretation may well be wrong
  - Minimal: minimal threshold for taking note of input
- Often, these thresholds are compared only to speech recognition score
  - In a more sophisticated system, one could imagine weighing together several factors such as contextual adequacy and common patterns of user behaviour

Selecting feedback: General principles

- Produce positive feedback only on highest possible level
  - For example, when giving positive feedback about contextual understanding, there is no need to also give feedback on perception
- Produce negative feedback only on the lowest level where a problem occurred
  - For example, if there is a problem with perception, the system should provide feedback on the perception level rather than e.g. the contextual interpretation level
- When producing negative feedback on some level, produce positive feedback on the level directly below it
  - For example, when indicating lack of contextual understanding, provide positive feedback och non-contextual understanding level

Feedback selection algorithm

- If ASR score below minimal threshold, select icm:per∗neg
- Else if system can find a relevant contextual interpretation:
  - If above high threshold, select no icm:und
  - Else if above medium threshold, select icm:und∗pos
  - Else if above low threshold, select icm:und∗chk
  - If above medium threshold:
    - If acceptable, integrate content and select icm:acc∗pos
    - Else select icm:acc∗neg
- Else if system can find a non-contextual interpretation:
  - Select icm:sem∗pos, icm:und∗neg
- Else if the system has heard something
  - If score above minimal threshold, select icm:per∗pos, icm:sem∗neg ("I heard you say '...'. I don't understand")
  - Else select icm:per∗neg ("I didn't hear you")
- Else do nothing, or wait *N* seconds then select icm:con∗neg ("Hello?")

Feedback on multiple levels

S: Lets see. What can I do for you?
U: (mumble)
S: Sorry, I didn't hear what you said. What can I do for you? [icm:per∗neg]
U: Channel one (ASR gets "channel no")
S: I heard you say 'channel no'. Sorry, I don't understand. What can I do for you? [icm:per∗pos, icm:sem∗neg]
U: Channel one
S: Channel one. I didn't quite understand. Do you want to add a program or change channel? [icm:sem∗pos, icm:und∗neg, ask]

Feedback on multiple levels

U: Turn off the t v
S: You want me to turn off the TV. Sorry, I cannot control the TV set. Do you want to add a program or change channel? [icm:und∗pos, icm:acc∗neg]
U: Pardon [icm:per∗neg]
S: Sorry, I cannot control the TV set. Do you want to add a program or change channel?
U: Change to channel four
S: Okay, Channel four. The channel has been changed. [icm:acc∗pos, icm:und∗pos, confirm]

Outline

Dealing with reactions to positive and checking feedback

- The system also needs to deal appropriately with the user reaction to feedback
  - Give appropriate response
  - Update the information state appropriately
- After checking feedback, (e.g. "channel one, is that correct?"):
  - If user gives no response or negative response, system assumes that its hypothesis was wrong;
  - If user gives positive response, system assumes it was right
- After positive feedback, (e.g. "channel one."):
  - If user gives no response or positive response, system continues to assume it was right;
  - If user gives negative response, system assumes it was wrong
- In IBiS, for an answer to be integrated it must be relevant to a question on ISSUES

Issue-based grounding

- Checking feedback on contextual understanding level explicitly raises *understanding-issue*
  - icm:und∗chk:C raises ?und(C)
  - Paraphrase: "Is C the meaning of the previous utterance?"
  - similar to Ginzburg's "content-question"
  - Positive response leads to adding C to shared commitments
  - Negative or no response leads to no action
- Positive understanding feedback does not raise the understanding-issue explicitly
  - In fact, tentatively assumes it has been positively resolved but makes it available for interpretation of short answers, in this case, "yes" or "no"
  - Negative response leads to retracting C
  - Positive or no response leads to no action
- So far, only implemented for understanding level

ISSUES, QUD and accommodation

- We distinguish QUD and ISSUES:
  - QUD (local): Questions available for resolution of short answers
  - ISSUES (global): Issues/questions that have been raised but not yet resolved
- Questions can drop off QUD but remain on ISSUES, e.g. if a question is not addressed withing a certain time span or a certain number of utterances
- Questions can be removed from ISSUES but remain on QUD, e.g. if a question is immediately resolved
  - Also, positive feedback implicitly makes understanding-question available for elliptical responses (pushed on QUD) but does not raise it (not on ISSUES)
- QUD-to-ISSUES accommodation
  - If an answer is given which does not address any question on ISSUES, but does address a question Q on QUD: accommodate Q to ISSUES (this amounts to raising Q)

Feedback, ISSUES and QUD

- Assume previous move had content C
  - Checking feedback
    - Push ?und(C) on ISSUES and QUD
  - Positive feedback
    - Add C to shared commitments
    - Push ?und(C) on QUD (but not on ISSUES)
    - "yes" or "no" $\Rightarrow$ do QUD-to-ISSUES accommodation
  - Integrating responses to understanding-issues
    - Done by a single rule, regardless of how the understanding-issue was raised
    - Given that ?und(C) is on ISSUES,
    - "yes" $\Rightarrow$ add C to shared commitments if not already there
    - "no" $\Rightarrow$ retract C from shared commitments if it's there (actually, backtrack to saved previous shared commitments)

Example: Positive response to checking feedback

S: Lets see. What channel do you want?
U: Channel six
S: Channel six, is that correct?

$$\left[\begin{array}{lll} \text{SHARED} & = & \left[\begin{array}{lll} \text{COM} & = & \{\ \ldots\ \} \\ \text{QUD} & = & \langle\ \text{?und(channel(6))}, \ldots\ \rangle \\ \text{ISSUES} & = & \langle\ \text{?und(channel(6))}, \ldots\ \rangle \end{array}\right] \end{array}\right]$$

U: Yes

Integrate user answer to understanding-question $\Rightarrow$ add content to COM
Downdate ISSUES
Downdate QUD

$$\left[\begin{array}{lll} \text{SHARED} & = & \left[\begin{array}{lll} \text{COM} & = & \{\ \ldots, \text{channel(6)}, \ldots\ \} \\ \text{QUD} & = & \langle\ \ldots\ \rangle \\ \text{ISSUES} & = & \langle\ \ldots\ \rangle \end{array}\right] \end{array}\right]$$

S: The channel has been changed.

Example: No response to checking feedback

S: Lets see. What channel do you want?

U: Channel six

S: Channel six, is that correct?

$$\left[\begin{array}{lll} \text{SHARED} & = & \left[\begin{array}{lll} \text{COM} & = & \{\ \ldots\ \} \\ \text{QUD} & = & \langle\ \text{?und(channel(6))}, \ldots\ \rangle \\ \text{ISSUES} & = & \langle\ \text{?und(channel(6))}, \ldots\ \rangle \end{array}\right] \end{array}\right]$$

U: (silent)

No followup to und-question $\Rightarrow$ Downdate ISSUES[1]

$$\left[\begin{array}{lll} \text{SHARED} & = & \left[\begin{array}{lll} \text{COM} & = & \{\ \ldots\ \} \\ \text{QUD} & = & \langle\ \ldots\ \rangle \\ \text{ISSUES} & = & \langle\ \ldots\ \rangle \end{array}\right] \end{array}\right]$$

Reraise question about channel

S: So, what channel do you want?

[1] We don't want to force user to answer by repeating the und-question; better to repeat original question instead.

Example: No response to positive feedback

S: Lets see. What channel do you want?
U: channel six
S: OK, channel six.

$$\left[\begin{array}{lll} \text{SHARED} & = & \left[\begin{array}{lll} \text{COM} & = & \{\ \ldots, \text{channel}(6), \ldots\ \} \\ \text{QUD} & = & \langle\ \text{?und}(\text{channel}(6)), \ldots\ \rangle \\ \text{ISSUES} & = & \langle\ \ldots\ \rangle \end{array}\right] \end{array}\right]$$

U: (silent)

Downdate QUD

$$\left[\begin{array}{lll} \text{SHARED} & = & \left[\begin{array}{lll} \text{COM} & = & \{\ \ldots, \text{channel}(6), \ldots\ \} \\ \text{QUD} & = & \langle\ \ldots\ \rangle \\ \text{ISSUES} & = & \langle\ \ldots\ \rangle \end{array}\right] \end{array}\right]$$

S: The channel has been changed.

Example: Negative response to positive feedback

S: Lets see. What channel do you want?

U: channel six

S: OK, channel five.

$$\left[\begin{array}{ll} \textsc{shared} = & \left[\begin{array}{lll} \textsc{com} & = & \{\ \ldots, \text{channel}(5), \ldots\ \} \\ \textsc{qud} & = & \langle\ ?\text{und}(\text{channel}(5)), \ldots\ \rangle \\ \textsc{issues} & = & \langle\ \ldots\ \rangle \end{array}\right] \end{array}\right]$$

U: No

QUD-to-ISSUES accommodation $\Rightarrow$ push und-question on ISSUES

$$\left[\begin{array}{ll} \textsc{shared} = & \left[\begin{array}{lll} \textsc{com} & = & \{\ \ldots, \text{not}(\text{channel}(5)), \ldots\ \} \\ \textsc{qud} & = & \langle\ ?\text{und}(\text{channel}(5)), \ldots\ \rangle \\ \textsc{issues} & = & \langle\ ?\text{und}(\text{channel}(5)), \ldots\ \rangle \end{array}\right] \end{array}\right]$$

Integrate user answer $\Rightarrow$ add negated content to COM

Downdate ISSUES since answer is in COM

Downdate QUD

$$\left[\begin{array}{ll} \textsc{shared} = & \left[\begin{array}{lll} \textsc{com} & = & \{\ \ldots, \text{not}(\text{channel}(5)), \ldots\ \} \\ \textsc{qud} & = & \langle\ \ldots\ \rangle \\ \textsc{issues} & = & \langle\ \ldots\ \rangle \end{array}\right] \end{array}\right]$$

S: Not channel five. So, What channel do you want?

Correcting the system

S: Lets see. What can I do for you?
U: add program
S: Okay. Add a program.
S: Lets see. What channel do you want?
U: channel six
S: OK, channel five.
U: no
S: not channel five. So, What channel do you want?
U: channel six
S: Okay. channel six.

Outline

Sequencing ICM

- Discourse particles
  - "so" may be used to reraise a previously discussed question
  - "right", "anyway", "let's see..."
- Verbose sequencing
  - "Let's talk about ...", "Let's return to the issue of ..."
- How do we classify these?
- We correlate them with updates to the common ground

Sequencing moves generated by IBiS

- Formal notation schema:
  - icm:*Type*:*Content*
- Loading a dialogue plan
  - icm:loadplan - "let's see."
- Question accommodation
  - icm:accommodate - "alright."
- Reraising an issue
  - icm:reraise - "so, "
  - icm:reraise:Q - "returning to the issue of ..."

Example: ICM for reraising issue and for loading a plan

S: Welcome to the travel agency!
U: Price information
S: OK. Lets see. How do you want to travel? [icm:acc∗pos, icm:loadplan, ask]
U: By flight
S: OK. What city do you want to go to?
U: Bielefeld
S: OK. What city do you want to go from?
U: Do I need a visa ?
S: OK, you want to know about visa. Let's see. What country are you from?
U: Sweden
S: Okay. No, you don't need a Visa.
S: Returning to the issue of price. Lets see. So, what city do you want to go from? [icm:reraise:?visa, icm:loadplan, icm:reraise]

Example: Accommodation ICM

S: Lets see. What can I do for you?
U: Channel five today
S: Alright. You want to add a program. Lets see. Okay. channel five... today... What time do you want to start recording? [icm:accommodate, icm:und∗pos, icm:loadplan, icm:acc∗pos, icm:und∗pos:channel(5), icm:und∗pos;day(today), ask]
U: From six thirty to seven thirty
S: Okay. from six thirty... to seven thirty... The program has been added.

Outline

Conclusions

- By extending the range of ICM handled by systems, their communication becomes more natural and comprehensive
- We have provided a classification of feedback and sequencing ICM useful in a dialogue system, and implemented it
- Issue-based grounding provides mechanisms allowing the user to react to system feedback
- Sequencing moves can be correlated with updates to common ground, and used to signal these updates to the user

Related recent & current work

- Multimodal menu-based dialogue, including multimodal grounding and feedback
- Multimodal barge-in: user can interrupt system speech by clicking on a button, or by speaking
- Research on other functions of grounding and feedback
  - Apart from coordinating on information ("informational coordination"), feedback can also aid *semantic* coordination (coordinating on the meanings of linguistic expressions)

Possible future work

- Extend issue-based grounding to all levels
  - Add a field SHARED.NIM ("pending")
- Extend the range of ICM generated and interpreted by the system
- Improve CR (clarification request) coverage (Ginzburg & Cooper, Purver)
  - Currently, system can generate CRs on the (pragmatic) understanding level ("Bielefeld?") but these are typically generated in a more verbose manner ("Bielefeld, is that correct?") to avoid ambiguity
  - System does not understand CRs from user

Possible future work

- Incremental asynchronous grounding
  - Generate “mm”s while user is speaking; endpoint detection for fast turntaking (Skantze)
  - When the system is interrupted, how much of what it was saying can assumed to be grounded? Depends on how far it got, and whether the interruption was a relevant utterance.
- Complement current ad-hoc forms with variants based on corpus studies and existing research
  - ”discourse markers” (Schiffrin)
  - ”cue phrases” (Grosz & Sidner and others)