Dialogue management with linear logic: the role of metavariables in questions and clarifications

Vladislav Maraev, Jean-Philippe Bernardy and Jonathan Ginzburg

CLASP Seminar · Feb 24, 2021

Dialogue coherence

Task-oriented dialogue (from Gao et al. 2020):

U: Should we meet 2pm or 5pm?
S: Okay.
U: 2pm or 5pm? Just to confirm.
S: Okay.

Dialogue manager

Dialogue manager (DM) selects *appropriate* system actions depending on the current state and the external context.

Types:

- *hand-crafted* (Larsson, 2002; Jokinen, 2009)
- *statistical models* (Rieser and Lemon, 2011; Young et al., 2010)
- *end2end (no DM)* (Huang et al., 2020; Gao et al., 2020)

Theoretical models of dialogue

Only few DM approaches reflect advancements in dialogue theory.

- non-sentential utterances

  A: Does John live in Paris?
  B: from January.

- clarification requests and insertion sequences

  U: When is there a bus from Valand?
  S: Which number?
  U: 55.
  S: It is in 15 minutes.

What we did

- modular system with **domain-general rules**
- dialogue manager that goes hand in hand with the theory (KoS)
- rich information state, **ISU approach**
- a proof-search engine based on **linear logic**

Next...

Background

Demo

Questions

Clarification requests and follow-up questions

KoS-inspired dialogue management

Evaluation and summary

KoS: domain-general conversational relevance

In KoS (Ginzburg, 2012) language is compared to a game, containing players (interlocutors), goals and rules. KoS represents language interaction by a dynamically changing context. The meaning of an utterance is then how it changes the context.

- **Dialogue Game Board (DGB)** for each dialogue participant
  - Moves
  - QUD
  - ...
- based on Type Theory with Records (TTR) (Cooper, 2005)
- our previous implementation was based on intuitionistic version of TTR (Maraev et al., 2018).

Information-state update approach (ISU)

- **Rich** information-state which includes a hierarchy of facts: thought to be shared and not yet publicised.
- Information-state update approach (Larsson, 2002; Ginzburg, 2012)
- Example: *QUD-incrementation*, update current set of *QUD* if the latest utterance is a question.
- Supports wide range of CRs and follow-ups, and also contextually relevant contributions, such as over-answering.

Proof search

- axioms: *Leave* 55 *Valand* 11.50
- rules of inference:
  Leave x Valand y $\rightarrow$ *Arrive x CentralStationen* $(y + 45\ min)$
- metavariables (lowercase letters)

Linear logic

- hypotheses may be used only once:
  $IsAt\ x\ Valand\ y \multimap IsAt\ x\ CentralStationen\ (y + 45\ min)$
- unrestricted implication ($\rightarrow$) still remains, for immutable facts

$$Hungry\ Vlad \rightarrow Eat\ Vlad\ IceCream \multimap Happy\ Vlad$$

To our knowledge Dixon et al. (2009) were the first to advocate the use of linear logic for dialogue management and planning (but no domain-generality and ISU).

Linear logic as DM framework

- linear rule corresponds to an action of an agent
- actions realised as actual interactions constitute an observable dialogue, e.g.
  - sending message to the outside world (speech, movement etc.)
  - perceptory subsystem (events in the outside world lead to updates in IS)
- multiset of linear hypotheses represents the current information-state of the agent

Note: hypotheses have no hierarchy (unlike in TTR), but can be wrapped in constructors, e.g. *Unsure P* or *QUD Q*.

Next...

Background

Demo

Questions

Clarification requests and follow-up questions

KoS-inspired dialogue management

Evaluation and summary

Demo

Next...

Background

Demo

Questions

Clarification requests and follow-up questions

KoS-inspired dialogue management

Evaluation and summary

Question type

$A : \mathit{Type}$
$P : A \rightarrow \mathit{Prop}$

- $(P\ x)$ is the interpretation of short answer x as a proposition.
- The intent of the question is to find out about the value x of type A which makes $(P\ x)$ true.
- Question can be represented as $Q\ A\ x\ (P\ x)$

Question type

$$A : Type$$
$$P : A \rightarrow Prop$$

- $(P\ x)$ is the interpretation of short answer x as a proposition.
- The intent of the question is to find out about the value x of type A which makes $(P\ x)$ true.
- Question can be represented as $Q\ A\ x\ (P\ x)$

Example

A: Where does John live?

$$\begin{aligned} A &= Location \\ P\ x &= \lambda x.Live\ John\ x \\ [\![\ Paris\ ]\!] &= ShortAnswer\ Location\ Paris \end{aligned}$$

Wh-questions

A: Where does John live?
B: in London

$$QUD\ (Q\ Location\ x\ (Live\ John\ x))$$
$$ShortAnswer\ Location\ London$$

$$processShort : (a : Type) \rightarrow (x : a) \rightarrow (p : Prop) \rightarrow ShortAnswer\ a\ x \multimap QUD\ (Q\ a\ x\ p) \multimap p$$

- **metavariables** are declared via Π type binders, can be read as $\forall$a, $\forall$x etc.
- **relevance**: we demand that types in the answer and in the question match
- **unification**: x occurs in p and can be unified with *London*

Polar questions

A: Does John live in Paris
B: yes / no

$$
\begin{aligned}
A &= \mathit{Bool} \\
P\,x &= \lambda x.\mathbf{if}\ x\ \mathbf{then}\ (\mathit{Live}\ \mathit{John}\ \mathit{Paris}) \\
&\qquad\qquad \mathbf{else}\ \mathit{Not}\ (\mathit{Live}\ \mathit{John}\ \mathit{Paris}) \\
[\![\,\mathit{yes}\,]\!] &= \mathit{ShortAnswer}\ \mathit{Bool}\ \mathit{True}
\end{aligned}
$$

Polar questions

A: Does John live in Paris
B: yes / no

$$
\begin{aligned}
A &= \mathit{Bool} \\
P\,x &= \lambda x.\mathbf{if}\ x\ \mathbf{then}\ (\mathit{Live\ John\ Paris}) \\
&\qquad\qquad \mathbf{else}\ \mathit{Not}\ (\mathit{Live\ John\ Paris}) \\
[\![\,\mathit{yes}\,]\!] &= \mathit{ShortAnswer\ Bool\ True}
\end{aligned}
$$

- What about answers like “possibly”, “probably”, “since yesterday” etc.?

Polar questions: refined

Short answers can be adverbs (A = $Prop \rightarrow Prop$).

$$
\begin{aligned}
A &= Prop \rightarrow Prop \\
P &= \lambda m.m\ (Live\ John\ Paris) \\
[\![\ yes\ ]\!] &= ShortAnswer\ (Prop \rightarrow Prop)\ (\lambda p.p) \\
[\![\ no\ ]\!] &= ShortAnswer\ (Prop \rightarrow Prop)\ (\lambda p.Not\ p) \\
[\![\ from\ January\ ]\!] &= ShortAnswer\ (Prop \rightarrow Prop) \\
&\qquad (\lambda p.FromJanuary\ p)
\end{aligned}
$$

Bonus: laughing at questions

from Ginzburg et al. (2020):

Journalist: (smile)	Dreierkette auch ‘ne Option?
	(Is the three-in-the-back also an option?)
Manuel Neuer:	fuh fuh fuh
	(brief laugh)

$$A = Prop \rightarrow Prop$$
$$P = \lambda m.m\ IsOptionDreierkette$$
$$[\![fuh\ fuh\ fuh]\!] = ShortAnswer\ (Prop \rightarrow Prop)\ (\lambda x.Laughable\ x)$$

Negative polar questions

A: Doesn’t John like Bananas?
B: no / no he doesn’t / yes / qualifier, e.g. "after tennis"

$$
\begin{aligned}
Q\ Multi\ (\lambda x.\mathbf{case}\ x\ \mathbf{of}\ AmbiguousNo &\rightarrow Trivial \\
DefiniteNo &\rightarrow \neg\ (Like\ John\ Bananas) \\
DefiniteYes &\rightarrow Like\ John\ Bananas \\
Qualifier\ m &\rightarrow m\ (Like\ John\ Bananas))
\end{aligned}
$$

The meaning of short answers always depends on the context

Next...

Background

Demo

Questions

Clarification requests and follow-up questions

KoS-inspired dialogue management

Evaluation and summary

Metavariables in the information-state I

- "The mouse eats something"
  Eat Mouse x

Metavariables in the information-state II

- "The mouse eats spaghetti"
  Eat Mouse Spaghetti

Answering the question

$$(a : Type) \to (x : a) \to (p : Prop) \to$$
$$QUD\ (Q\ a\ x\ p) \to p \multimap ShortAnswer\ a\ x$$

- Note: $A \multimap B$ is a syntactic sugar for $A \multimap (A \otimes B)$.

Intuition: we might not want to answer the question if x not known, or if x is ambiguous (otherwise the answer would be non-resolving).

Unique and concrete type-former: $(x : A) \to_! B$

The rule $(x : A) \to_! B$ introduces the metavariable x, but can only fire when:

- x is **made ground** (it is bound to a term which does not contain any metavariable)
- x is **unique**

Unique and concrete type-former: $(x : A) \to_! B$

The rule $(x : A) \to_! B$ introduces the metavariable x, but can only fire when:

- x is **made ground** (it is bound to a term which does not contain any metavariable)
- x is **unique**

Thus, we can produce and answer only in such case:

$$\begin{aligned} &\mathit{produceAnswer} : (a : \mathit{Type}) \to (x : a) \to_! (p : \mathit{Prop}) \to \\ &\quad \mathit{QUD}\ (Q\ a\ x\ p) \to p \rightarrowtriangle \mathit{ShortAnswer}\ a\ x \end{aligned}$$

Clarification requests

Consider the question “What is being eaten?” represented as $Q\ x\ (Eat\ y\ x)$), with the state:

Eat John Mars
Eat Mary Mars

Mars!

Clarification requests

Consider the question “What is being eaten?” represented as $Q\ x\ (Eat\ y\ x)$), with the state:

Eat John Mars
Eat Mary Mars

Mars!

Eat John Mars
Eat Mary Twix

By whom?

Issuing CR

$$[a : Type; x : a; p : Prop; qud :: QUD\ (Q\ x\ p); proof :: p] \rightarrow_? CR$$

- we leave the exact form of CR unspecified (can be domain-specific)
- $\rightarrow_?$ operator test left-hand side to be non-unique or not fully ground

Implementing CR

$$
\begin{aligned}
&\mathit{Eat\ John\ Mars}\\
&\mathit{Eat\ Mary\ Twix}\\
&\mathit{ori} :: \mathit{QUD}\ (\mathit{Q\ Food}\ x\ (\mathit{Eat}\ y\ x))\\
&\mathit{cr}\ \ :: \mathit{QUD}\ (\mathit{Q\ Person}\ z\ (z = y))\\
&a\ \ \ :: \mathit{ShortAnswer\ Person\ Mary}
\end{aligned}
$$

after applying *processShort*:

$$
\begin{aligned}
&\mathit{Eat\ John\ Mars}\\
&\mathit{Eat\ Mary\ Twix}\\
&\mathit{ori} :: \mathit{QUD}\ (\mathit{Q\ Food}\ x\ (\mathit{Eat}\ y\ x))\\
&r\ \ \ :: \mathit{Mary} = y
\end{aligned}
$$

The original question becomes (by unification)
$\mathit{Q\ Food}\ x\ (\mathit{Eat\ Mary}\ x)$, and then can be unambiguously answered.

Further notes

1. The logical form of the question (z such that $z = y$) is typically realised in a complicated way.
2. In practice, the form of clarification questions will greatly vary depending on the context (Purver, 2004).
3. Answers could simply be exhaustive ("Mars or Twix"). In practice there can be an ambiguity threshold (e.g. answer is longer than n) after which clarification requests are preferred.

Paris, Denmark

Solution 1: extra arguments

$$Q\ City\ x\ (Live\ John\ x\ y)$$

Metavariable y can remain free for the duration of the dialogue. If answering the question demands clarification, this can be done using the mechanisms described above.

$$Live\ John\ Paris\ y$$

...but which Paris do you mean?

Summary

In sum, we leverage a feature of linear-logic proof search: **at any point in the scenario, the context can refer to metavariables**. In a dialogue application, metavariables represent a certain amount of flexibility in the scenario: *so far* the scenario works for any value which could be assigned to the metavariable. This means that at a further point the metavariable can be instantiated to some other value.

Next...

Background

Demo

Questions

Clarification requests and follow-up questions

KoS-inspired dialogue management

Evaluation and summary

Architecture

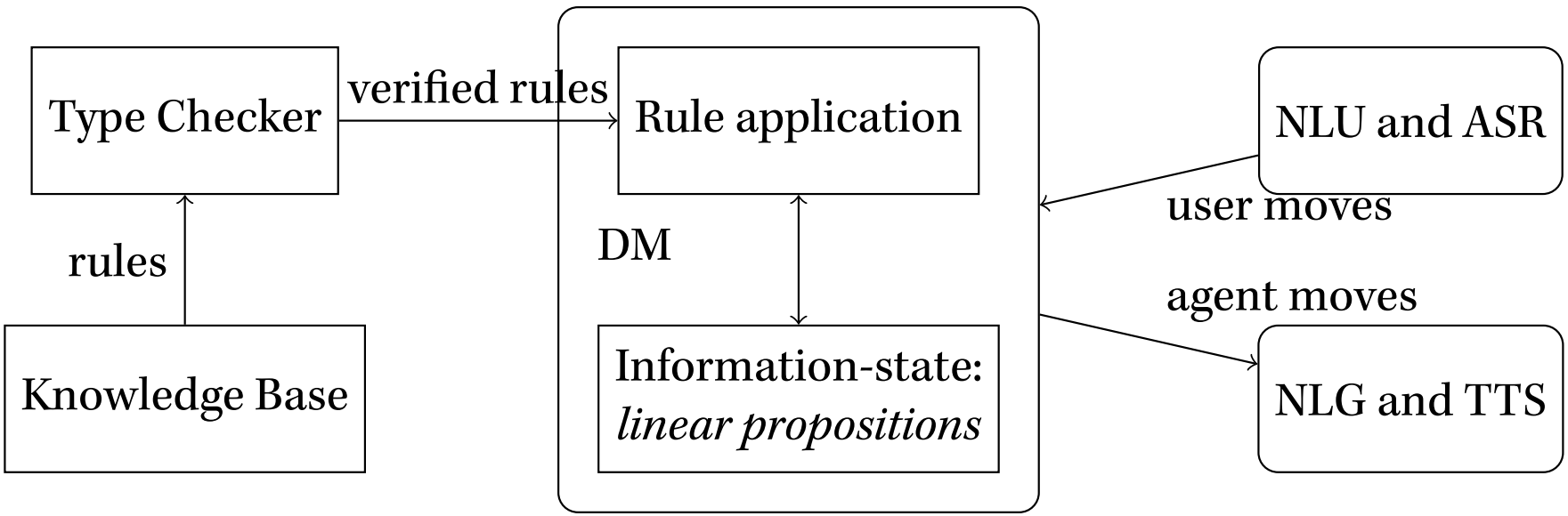

Figure: Architecture of a spoken dialogue system with a dialogue manager based on a linear logic framework.

Preliminaries I

5 types of moves:

Greet	*spkr addr*
CounterGreet	*spkr addr*
Ask	*question spkr addr*
ShortAnswer	*vtype v spkr addr*
Assert	*p spkr addr*

I/O:

- **Hearing**, e.g. $\mathit{Heard}\ (\mathit{Greet}\ S\ A)$, comes from external source (no rule needs to be fired)
- **Uttering**, e.g. $\mathit{Agenda}\ (\mathit{CounterGreet}\ A\ S)$ can be placed in information-state by some rule.

Preliminaries II

Pending and Moves

All the moves are recorded in *Moves* stack after they have been processed. We use hypothesis (*Pending m*) for move *m* which one has yet to actively react to.

Initial state

(domain-specific example)

_ :: *QUD Nil*; _ :: *Agenda Nil*; _ :: *Moves Nil*;

Back to demo

Next...

Background

Demo

Questions

Clarification requests and follow-up questions

KoS-inspired dialogue management

Evaluation and summary

Caveats and future work

1. Correction moves
2. Clarification of predicates
3. Semantic dependency between questions
4. Clarification of the *type* of metavariable
5. Probabilistic rules and probabilistic meanings
6. Incremental processing
7. Grounding

Benchmark

In evaluation we rely on the work by Ginzburg and Fernández (2010), who proposed a series of benchmarks for comparing different approaches to developing dialogue systems. See our paper for the results.

$$\mathit{ThatsIt} \multimap (\mathit{Thanking} \otimes \mathit{QA})$$

References I

Cooper, R. (2005). Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Dixon, L., Smaill, A., and Tsang, T. (2009). Plans, actions and dialogues using linear logic. *Journal of Logic, Language and Information*, 18(2):251–289.

Gao, J., Peng, B., Li, C., Li, J., Shayandeh, S., Liden, L., and Shum, H.-Y. (2020). Robust conversational ai with grounded text generation. *arXiv preprint arXiv:2009.03457*.

Ginzburg, J. (2012). *The Interactive Stance*. Oxford University Press.

Ginzburg, J. and Fernández, R. (2010). Computational models of dialogue. *The Handbook of Computational Linguistics and Natural Language Processing*, 57:1.

Ginzburg, J., Mazzocconi, C., and Tian, Y. (2020). Laughter as language. *Glossa: a journal of general linguistics*, 5(1).

References II

Huang, M., Zhu, X., and Gao, J. (2020). Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Jokinen, K. (2009). *Constructive dialogue modelling: Speech interaction and rational agents*, volume 10. John Wiley & Sons.

Larsson, S. (2002). *Issue-based dialogue management*. PhD thesis, University of Gothenburg.

Maraev, V., Ginzburg, J., Larsson, S., Tian, Y., and Bernardy, J.-P. (2018). Towards kos/ttr-based proof-theoretic dialogue management. In *Proceedings of the 22nd Workshop on the Semantics and Pragmatics of Dialogue*, Aix-en-Provence, France. SEMDIAL.

Purver, M. R. J. (2004). *The theory and use of clarification requests in dialogue*. PhD thesis, University of London.

References III

Rieser, V. and Lemon, O. (2011). *Reinforcement learning for adaptive dialogue systems: a data-driven methodology for dialogue management and natural language generation*. Springer Science & Business Media.

Young, S., Gašic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., and Yu, K. (2010). The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.