

Self Awareness for Better Common Ground

Ondřej Plátek and Filip Jurčiček

Charles University in Prague

Faculty of Mathematics and Physics

Institute of Formal and Applied Linguistics

{oplatek, jurcicek}@ufal.mff.cuni.cz

Abstract

Having set up several dialogue systems for multiple domains we realized that we need to introduce each system thoroughly to the users in order to avoid misunderstanding. Even with well-described systems, users still tend to request out-of-domain information and are often confused by the system response. In our ongoing work, we try to address these issues by allowing our conversational agent to speak about its abilities. Our agent is able to simply describe what it understands and why it decided to perform one of its summary actions. In this short paper, we present our current system architecture.

1 Introduction

Searching for a common ground in spoken conversation is a well-studied process known as *grounding* (Traum, 1994). We argue that out-of-domain questions are a-type of misunderstanding where the user incorrectly assumes a capability of the system. As a result, we think that dialogue systems should inform the user better about what their communication possibilities are.

Research in dialogue systems has focused so far on improving goal-oriented narrow domain systems. Such systems unfortunately communicate only about their domain and users easily request

out of domain information. The systems often cannot help the user if the user is lost in the conversation because it is hard to come up with correct response in out of domain context.

In our effort to develop conversational systems which try to learn domain knowledge from users, we realized that the first thing that users are trying to teach the system is to communicate better. One of our goals is to develop a policy which allows the system to confirm if the misunderstanding occurred, and identify its source. We attribute the misunderstanding to one of the system components language understanding (LU), dialogue management (DM) or natural language generation (NLG). Therefore we would like to confirm if the system understood the user badly, chose surprising action or used wrong words.

The first step is to present the user which actions the system is able to take and how it represents the dialogue history. In our system, we explicitly store information about the system, so the agent can communicate about its abilities. On the other hand, we do not attempt to explain to our users which facts the system is able to communicate about, and which facts are not present in its knowledge base. Consider example conversation between user *U* and system *S* to see how information about action *ask*, *slot date*, and *LU* component can help the user understand the system behaviour.

U: *Can you make a reservation for tonight evening at 20:00?*

S: *What movie do you want to see?*

U: *edge of tomorrow*

S: *Do you want to change the reservation for tomorrow?*

U: *No! What?*

S: *I asked you about the date because you change it by saying tomorrow. Do I understand you incorrectly?*

U: *Yes, I told you the name of the movie which is Edge of Tomorrow.*

It is hard for the user to imagine what she or

This research was partly funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221, core research funding, grant GAUK 227007, and also partially supported by SVV project number 260 224. This work has been using language resources distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2010013). Cloud computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144.

he should say to make the system understand better. We suggest that system should take initiative because the system is typically the interlocutor that misunderstands first.

2 Prototype description

Our prototype is text to text dialogue system with simple language understanding (LU) and natural language generation (NLG) component. The system updates a belief state similar to (Dušek et al., 2014) and it select actions based on handcrafted policy.

We implemented summary action which roughly correspond to core speech acts from (Traum and Hinkelman, 1992). Each summary action generates multiple candidates with different arguments taken from belief state¹. In addition to arguments, we explicitly store human-understandable *why features*² which should explain to user why this action is good candidate.

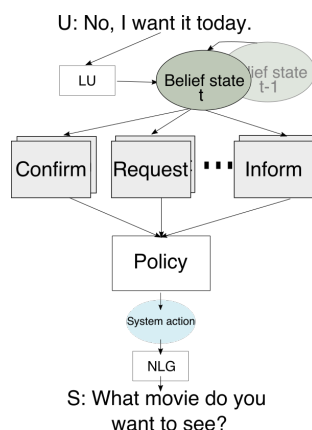


Figure 1: Architecture of our system.

The *why features* together with the system description from our knowledge base (KB) are used when the system explains the user what it tried to achieve. Our KB in addition to typical domain knowledge stores information that: a) Our system selects one from actions *Inform*, *Confirm*, *Request*, *Ask*, *Reject*, *Hello*, and *Good Bye* with corresponding arguments. b) Updates what user said in internal state.

The system also dynamically updates KB about the dialogue history so the system can use the KB as the only source of information. Having such dynamic KB allows the system not only to talk about

given domain, but also about its actions, and the dialogue history without additional summary actions.

3 Discussion and Future work

To our knowledge, recovering from miscommunication received a little attention (Skantze, 2007) in dialogue systems. We argue that the system initiative to explain its action is a sensible strategy for establishing a common ground (Traum, 1994) in situation where both interlocutors are lost.

We have extended standard dialogue system architecture so the resulting system has a chance to describe its behaviour. The key addition are so called *why features*³, the reasons for system actions. We also treat information about the system as another domain information in our KB and we do not need to implement special actions to handle help and error situations.

4 Conclusion

We presented a dialogue system architecture which allows to discuss the system behaviour easily, and thus better recovers from misunderstandings. We argued that self awareness of a system is beneficial. We plan to evaluate the system as future work.

References

- Ondrej Dušek, Ondrej Plátek, Lukáš Žilka, and Filip Jurčicek. 2014. Alex: Bootstrapping a spoken dialogue system for a new domain by real users. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 79.
- Gabriel Skantze. 2007. *Error handling in spoken dialogue systems: managing uncertainty, grounding and miscommunication*. Ph.D. thesis, Datavetenskap och kommunikation, Stockholm,.
- David R Traum and Elizabeth A Hinkelman. 1992. Conversation acts in task-oriented spoken dialogue. *Computational intelligence*, 8(3):575–599.
- David R. Traum. 1994. A Computational Theory of Grounding in Natural Language Conversation. Technical report, December.

¹Confirm(movie=Tomorrow), Confirm(movie="The")

²The policy does not decide based on the *why features*.

³The system action *confirm(date=tomorrow)* from the example conversation is stored with *why features* [user_said=tomorrow, old_value=today].