Interactional dynamics and the emergence of language games
Eshghi, Arash; Shalyminov, Igor; Lemon, Oliver

Published in:
CEUR Workshop Proceedings

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Interactional Dynamics and the Emergence of Language Games

Arash Eshghi
Interaction Lab
Heriot-Watt University
eshghi.a@gmail.com

Igor Shalyminov
Interaction Lab
Heriot-Watt University
o.lemon@hw.ac.uk

Oliver Lemon
Interaction Lab
Heriot-Watt University
o.lemon@hw.ac.uk

Abstract

Meaning is highly activity-specific, in that the action that a particular sequence of words is taken to perform is severely underdetermined in the absence of an overarching activity, or a ‘language-game’. In this paper, we combine a formal, incremental model of interactional dynamics and contextual update - Dynamic Syntax and Type Theory with Records (DS-TTR) - with Reinforcement Learning for word selection. We show, using an implemented system, that trial and error generation with a DS-TTR lexicon – a process we have dubbed babbling – leads to particular domain-specific dialogue acts to be learned and routinised over time; and thus that higher level dialogue structures - or how actions fit together to form a coherent whole - can be learned in this fashion. This method therefore allows incremental dialogue systems to be automatically bootstrapped from small amounts of unannotated dialogue transcripts, yet capturing a combinatorially large number of interactional variations. Even when the system is trained from only a single dialogue, we show that it supports over 8000 new dialogues in the same domain. This generalisation property results from the structural knowledge and constraints present within the grammar, and highlights limitations of recent state-of-the-art systems that are built using machine learning techniques only.

1 Introduction

Meaning is highly activity-specific, in that the action that a particular sequence of words is taken to perform, together with any perlocutionary effect that action might give rise to, is severely underdetermined in the absence of a particular overarching activity, or a ‘language-game’. Wittgenstein famously argued that the structure of a language-game, or how actions fit together to form a coherent whole, is irreducible. Arguably, this is the most unyielding obstacle facing not only theoretical approaches to pragmatics, but also dialogue system developers today. This suggests that particular dialogue structures are emergent, learned, and very frequently adjusted during interaction (Mills and Gregoromichelaki, 2010; Mills, 2011; Healey, 2008; Larsson and Cooper, 2008).

Despite this, recent and ongoing work in formal dialogue modelling suggests that not only language processing mechanisms, but also certain basic principles of contextual dynamics in dialogue do generalise across domains (Ginzburg, 2012; Kempson et al., 2016; Eshghi et al., 2015; Kempson et al., 2015; Purver et al., 2010). Even in a simple domain, there’s a lot of interactional variation that does not ultimately affect the overall communicative goal of a dialogue. For example, the dialogues in Fig. 1 (specifically the top two rows, where the lexicon is held constant) all lead to a context in which the user wants to buy a phone by LG. These dialogues can be said to be pragmatically synonymous for this domain. Arguably, a good model of interactional dynamics should be able to capture this synonymy.

In this paper, we show, using an implemented system (Kalatzis et al., 2016), that given Dynamic Syntax and Type Theory with Records (DS-TTR) (Kempson et al., 2001; Eshghi et al., 2012; Eshghi et al., 2015) as a low-level, incremental model of interactional and contextual dynamics, one can see dialogue acts, together with their associated local dialogue structures and procedural conventions as emergent and learned from interaction; and thus
that fully incremental dialogue systems can be bootstrapped from raw, unannotated example successful dialogues within a particular domain.

The model we present below combines DS-TTR with Reinforcement Learning for incremental word selection, where dialogue management and language generation are treated as one and the same decision/optimisation problem, and where the corresponding Markov Decision Process is automatically constructed. Using our implemented system, we demonstrate that using this system one can generalise from very small amounts of raw dialogue data, to a combinatorially large space of interactional variations, including phenomena such as question-answer pairs, over-answering, self- and other-corrections, split-utterances, and clarification interaction, when most of these are not even observed in the original data (see section 4.1).

1.1 Dimensions of Pragmatic Synonymy

There are two important dimensions along which dialogues can vary, but nevertheless, lead to very similar final contexts: interactional, and lexical. Interactional synonymy is analogous to syntactic synonymy - when two distinct sentences are parsed to identical logical forms - except that it occurs not only at the level of a single sentence, but at the dialogue or discourse level - Fig. 1 shows examples. Importantly as we shall show, this type of synonymy can be captured by grammars/models of dialogue context.

Lexical synonymy relations, on the other hand, hold among utterances, or dialogues, when different words (or sequences of words) express meanings that are sufficiently similar in a particular domain or activity - see Fig 1. Unlike syntactic/interactional synonymy relations, lexical ones can often break down when one moves to another domain: lexical synonymy relations are domain specific. Here we do not focus on these, but merely note that lexical synonymy relations can be captured using Distributional Methods (see e.g. Lewis & Steedman (2013)), or methods akin to Eshghi & Lemon (2014) by grounding domain-general semantics into the non-linguistic actions within a domain.

2 Dynamic Syntax (DS) and Type Theory with Records (TTR)

Dynamic Syntax (DS) a is a word-by-word incremental semantic parser/generator, based around the Dynamic Syntax (DS) grammar framework (Cann et al., 2005) especially suited to the fragmentary and highly contextual nature of dialogue. In DS, words are conditional actions - semantic updates; and dialogue is modelled as the interactive and incremental construction of contextual and semantic representations (Eshghi et al., 2015) - see Fig. 2. The contextual representations afforded by DS are of the fine-grained semantic content that is jointly negotiated/agreed upon by the interlocutors, as a result of processing questions and answers, clarification requests, acceptances, self-/other-corrections etc. The upshot of this is that using DS, we can not only track the semantic content of some current turn as it is being constructed (parsed or generated) word by word, but also the context of the conversation as whole, with the latter also encoding the grounded/agreed content of the conversation (see e.g. Fig. 2, and see Eshghi et al. (2015); Purver et al. (2010) for details of the model). Crucially for our model below, the inherent incrementality of DS together with the word-level, as well as cross-turn, parsing constraints it provides, enables the word-by-word exploration of the space of grammatical dialogues,
and the semantic and contextual representations that result from them.

These representations are Record Types (RT, see Fig. 2) of Type Theory with Records (TTR, (Cooper, 2005)), useful for incremental specification of utterance content, underspecification, as well as richer representations of the dialogue context (Purver et al., 2010; Purver et al., 2011; Eshghi et al., 2012). For reasons of lack of space, we only note that the TTR calculus provides, in addition to other operations, the subtype checking operation, ⊑, among Record Types (RT), and that of the Maximally specific Common Subtype (MCS) of two RTs, which both turn out to be crucial for the automatic construction of our MDP model, and feature checking (for more detail on the DS-TTR Hybrid, see (Espghi et al., 2012; Hough and Purver, 2014)).

3 The overall BABBLE method

We start with two resources: a) a DS-TTR parser DS (either learned from data (Eshghi et al., 2013), or constructed by hand), for incremental language processing, but also, more generally, for tracking the context of the dialogue using Eshghi et al.’s model of feedback (Eshghi et al., 2013; Eshghi, 2015; Eshghi et al., 2012). For reasons of lack of space, we only note that the TTR calculus provides, in addition to other operations, the subtype checking operation, ⊑, among Record Types (RT), and that of the Maximally specific Common Subtype (MCS) of two RTs, which both turn out to be crucial for the automatic construction of our MDP model, and feature checking (for more detail on the DS-TTR Hybrid, see (Espghi et al., 2012; Hough and Purver, 2014)).

4 Discussion

We have so far induced two prototype dialogue systems, one in an ‘electronic shopping’ domain (as exemplified by the dialogues in Fig. 1) and another in a ‘restaurant-search’ domain showing that incremental dialogue systems can be automatically created from small amounts of dialogue transcripts - in this case both systems were induced from a single successful example dialogue.

From a theoretical point of view, this shows that DS-TTR as an incremental model of interactional dynamics, with a domain-specific reward signal/goal is sufficient for certain word sequences becoming routinised and learned as ways of performing specific kinds of speech act within the domain, without any prior, procedural specifications of such actions. Thus, a dialogue system learns not only what it needs to do, but also how and when to do it (e.g. in a ‘restaurant-booking’ task, it learns to ask “What kind of cuisine would you like?”), in a situation where the user says she wants to book a table, but does not provide information about restaurant type: higher-, discourse-level dialogue structure is emergent from interaction in such a setting.

From the practical point of view of dialogue system development, the major benefits of this approach are in (1) more naturally interactive dialogue systems as the resulting systems are incremental and are thus able to handle inherently in-
only 1 example dialogue based on training from number of dialogue variants that can be processed the method implemented, a lower-bound on the example training dialogue.

dialogues that can be processed based on only 1 system can handle. This would again lead to a large number of variants that syntactic and lexical variations here, which possible after training the system on this dialogue. Note that we do not take into account possible syntactic and lexical variations on the above dialogue which can be parsed/generated by DS, and are thus automatically supported after training the system on this dialogue. Note that we do not take into account possible syntactic and lexical variations here, which would again lead to a large number of variants that the system can handle.

can lead to the same dialogue contexts being created, if we also consider similar variations in the two system processes, continuation, etc), then we arrive at a lower bound for the number of variations on the training dialogue of 8,192.

This remarkable generative power is due to the generalisation power of the DS grammar, combined with the system’s DM/NLG policy which is created by searching through the space of possible (successful) dialogue variants.

5 Conclusion and ongoing work

We show how incremental dialogue systems can be automatically learned from example successful dialogues in a domain, with Dialogue Acts and discourse structure emergent rather specified in advance. This method allows rapid domain transfer – simply collect some example (successful) dialogues in a ‘slot-filling’ domain, and retrain. At present this is fully automated, and only requires checking that the DS lexicon covers the input data. We are currently applying this method to the problem of learning (visual) word meanings (groundings) from interaction.

Figure 2: Semantics to MDP state encoding with RT features

<table>
<thead>
<tr>
<th>Grounded Semantics ($c_g$)</th>
<th>Pending Semantics ($c_p$)</th>
<th>Dialogue so far</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2$ : $e$</td>
<td>$x_2$ : $e$</td>
<td>SYS: What would you like?</td>
</tr>
<tr>
<td>$x_{\text{like}}$ : $e$</td>
<td>$x_{\text{like}}$ : $es$</td>
<td>USR: a phone</td>
</tr>
<tr>
<td>$x_{\text{brand}}$ : $e$</td>
<td>$x_{\text{brand}}$ : $e$</td>
<td>SYS: by which brand?</td>
</tr>
<tr>
<td>$p_{\text{pre}}(2)$ : $t$</td>
<td>$p_{\text{pre}}(2)$ : $t$</td>
<td>USR: by Apple</td>
</tr>
<tr>
<td>$p_{\text{subject}}(2)$ : $t$</td>
<td>$p_{\text{subject}}(2)$ : $t$</td>
<td>USR: a phone</td>
</tr>
<tr>
<td>$p_{\text{object}}(2)$ : $t$</td>
<td>$p_{\text{object}}(2)$ : $t$</td>
<td>SYS: by which brand?</td>
</tr>
</tbody>
</table>

RT Feature ($\phi$): |

$\begin{bmatrix}
  x_{10} & : & e & | & x_{10} & : & e & | & x_{10} & : & e \\
p_{15}(\text{brand}(10)) & : & t & | & p_{15}(\text{brand}(10)) & : & t & | & p_{15}(\text{brand}(10)) & : & t \\
p_{2}(\text{pre}(3)) & : & t & | & p_{2}(\text{pre}(3)) & : & t & | & p_{2}(\text{pre}(3)) & : & t \\
\end{bmatrix}$

State: \[
\begin{bmatrix}
F_1 & \downarrow & F_2 & \downarrow & F_3 & \downarrow & F_4 & \downarrow & F_5 & \downarrow \\
1 & & 1 & & 1 & & 1 & & 1 & \\
0 & & 1 & & 0 & & 1 & & 1 & \\
\end{bmatrix}
\]
References


