Dialogue State Tracking Using Stack Transducers

Suna Bensch

Department of Computing Science, Umeå University
suna@cs.umu.se

Abstract

This position paper presents our initial ideas on using stack transducers for dialogue state tracking. Stack transducers are abstract devices that translate an input to an output while being able to access their stacks. We consider a representation where user intention and system action are input and output to the stack transducer and the dialogue history is stored in the stack and accessible. We argue that this representation is highly efficient both in the sense of descriptional size, and complexity of implementation. This work is still at an early experimental stage and we present our ideas that seem acceptable at this point of analysis.

1. Introduction

Spoken dialogue systems are composed of multiple complex components, such as speech recognition, semantic analysis, dialogue management, and speech synthesis. Dialog management is responsible for *tracking the states* of a conversation, in particular, for deciding on the next appropriate step (e.g. verbal utterance). It has been shown that dialogue state tracking (also referred to as *Belief tracking*) compensates for speech recognition errors and reduces ambiguity (Henderson et al., 2014; Henderson et al., 2014; Thomson and Young, 2010). Dialogue state tracking is usually based on user intention, current dialogue state, dialogue history, context, and purpose of a dialogue. Dialogue states are often represented as a collection of slot-value pairs that represent user intentions. For instance, a dialogue system for flight reservation might have slots for city of departure, city of destination, etc. User intentions are often inferred from utterances using shallow parsing techniques (e.g. Named-Entity Recognition (NER), semantic role assignment (Sutherland et al., 2015), etc.). As the dialogue proceeds, turn for turn, dialogue state tracking stores and updates the dialogue states.

2. Background

Approaches for development of dialogue management systems are divided into knowledge-based dialogue management (i.e. hand-crafted finite-state and planning approaches) and data-driven approaches (Lee et al., 2010). Recent hybrid approaches to dialog management combine the benefits of knowledge-based and data-driven approaches (Lison, 2015; Ramachandran and Ratnaparkhi, 2015). Most systems today maintain a probability distribution over possible dialogue states and use slot-value pairs representation (Mehta et al., 2010; Wang and Lemon, 2013).

The authors in (Ramachandran and Ratnaparkhi, 2015) represent a user intention as relational tree, rather than a slot-value pair. A relational tree is similar to a dependency graph but constructed between entities from a NER. A user intention represented as a relational tree is extracted from each user utterance and as the dialogue proceeds relational trees are stacked on top of each other. Thus, stacked relational trees correspond to dialogue states which, in addition to modelling user intention, also model the conversational focus and the structure of subdialogues. The authors in (Ramachandran and Ratnaparkhi, 2015) expand their deterministic model of dialog state tracking to a probability distribution over stacked relational trees.

In (Hori et al., 2009; Hori et al., 2009; Hurtado et al., 2010) dialogue state tracking and the next dialogue step generation is realised with finite-state transducers. A finite-state transducer is an abstract device which translates an input to an output. Finite-state transducers are frequently used in natural language processing, for example, in speech processing (Mohri et al., 2008), and machine translation (Graehl et al., 2008).

The authors in (Hori et al., 2009; Hori et al., 2009; Hurtado et al., 2010) use a probabilistic approach to dialogue state tracking. The input to the finite-state transducer is the user intention represented in slot-value format and the output of the finite-state transducers are system actions.

3. Stack Transducers

We propose to use stack transducers, that were introduced in (Bensch et al., 2016), to track dialogue states during a conversation. A stack transducer is a finite-state transducer equipped with a stack, whose contents can be accessed, but not altered, with a stack pointer.

There are two modes in which a stack transducer operates. In digging mode, the stack transducer can produce output while reading an input and the stack pointer is inside the stack. In non-digging mode, the stack transducer can only produce output when the stack pointer is pointing on the topmost stack cell, but it can read input while the stack pointer is inside the stack.

4. Future Work

As a first step, abstracting from the representation of user intention (e.g. slot-value pairs, structured representation), we will formalise dialogue structure in terms of dialogue state tracking using stack transducers, so as to get a theoretical fundament of dialogue structure. The input to the stack transducer is assumed to be abstract representations of user intentions, the output is assumed to be system actions, and the stack stores the dialogue history.

The stack provides information of how, and how much, of the dialogue history has to be processed in order to be able to trigger system actions (i.e. emit output). Anaphor resolution is a particular problem where accessing the stack contents can be of advantage. For instance, entities introduced in the beginning of a dialogue can be accessed and reacted upon during the proceeded dialogue.

Since tree or graph representations of user intention provide insight to deeper structural relations than slot-value pairs, we will explore finite-state transducers whose input and output are trees or graph respresentations of user intentions and system actions, respectively.

In further steps, we will train our hybrid model on dialogue corpora. Data-driven approaches are very promising but require a large number of dialogue corpora. Hybrid approaches to dialogue modelling aim at constraining the set of possible actions with conventional rules (Lee et al., 2010; Williams, 2008) in order to incorporate domain knowledge or context, but are still learnable from data, compensate for speech recognition errors and model ambiguous situtations.

References

S. Bensch and J. Björklund and M. Kutrib. 2016. Deterministic Stack Transducers. *Proceedings of the 21st International Conference on Implementation and Application of Automata*, pages 27–38.

J. Graehl and K. Knight and J. May. 2008. Training tree transducers. *Computational Linguistics* (34), pages 391–427.

H. Henderson, B. Thomason, and J. Williams. 2014. The third dialog state tracking challenge. *Proceedings of IEEE Spoken Language Technology*.

H. Henderson, B. Thomason, and J. Williams. 2014. The second dialog state tracking challenge. *Proceedings of the SIGDIAL 2014 Conference*, page 263.

C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009. Statistical Dialog Management Applied to WFST-Based Dialog Systems. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009)*, pages 4793–4796.

C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009. Weighted Finite State Transducer Based Statistical Dialog Management. *IEEE Workshop on Automatic Speech Recognition & Understanding, 2009 (ASRU 2009)*, pages 490–495.

L. Hurtado, J. Planells, E. Segarra, E. Sanchis and D. Griol. 2010. A Stochastic Finite-State Transducer Approach to Spoken Dialog Management. *Proceedings of the 11th Annual Conference of the International Conference Speech Communication Association 2010 (INTERSPEECH 2010)*, pages 3002–3005.

C. Lee, S. Jung, K. Kim, D. Lee and G.G. Lee. 2010. Recent Approaches to Dialog Management for Spoken Dialog Systems. *Journal of Computing Science and Engineering*, 4(1): 1–22.

P. Lison. 2015. A hybrid approach to dialogue management based on probabilistic rules. *Computer, Speech and Language*, 34(1): 232–255.

N. Mehta, R. Gupta, A. Raux, D. Ramachandran and S. Krawczyk. 2010. Probabilistic Ontology Trees for Belief Tracking in Dialog Systems. *Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialgoue*, pages 37–46. 2010 Association for Computational Linguistics.

M. Mohri, F.C.N. Pereira and M. Riley. 2008. In: Rabiner, L., Juang, F. (eds.) Handbook on speech processing and speech communication, Part E: Speech recognition. Springer 2008.

D. Ramachandran and A. Ratnaparkhi. 2015. Belief Tracking with Stacked Relational Trees. *Proceedings of the SIGDIAL 2015 Conference*, pages 68–76. 2015 Association for Computational Linguistics.

A. Sutherland, S. Bensch and T. Hellström. 2015. Inferring Robot Actions from Verbal Commands Using Shallow Semantic Parsing. *International Conference on Artificial Intelligence 2015 (ICAI-2015)*, pages 28–34.

B. Thomson and SJ. Young. 2010. Bayesian update of dialogue state: A POMP framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588.

Z. Wang and O. Lemon. 2013. A Simple and Generic Belief Tracking Mechanism for the Dialog State Tracking Challenge: On the believability of observed information. *Proceedings of SIGDIAL 2013*, pages 423–432. 2013 Association for Computational Linguistics.

J.D. Williams. 2008. The best of both worlds: Unifying conventional dialog systems and POMDPs. *Proceedings of the Annual International Conference Speech Communication*, pages 1173–1176.