Intelligent Personal Assistants

Sebastian Springenberg

June 22, 2016

Speech Technology Seminar, Institute of Computer Science Hamburg

Table of contents

Introduction

Motivation

What do you expect from an intelligent personal assistant?

Motivation -cont.-

Say what you want, let the system infer the best course of action.

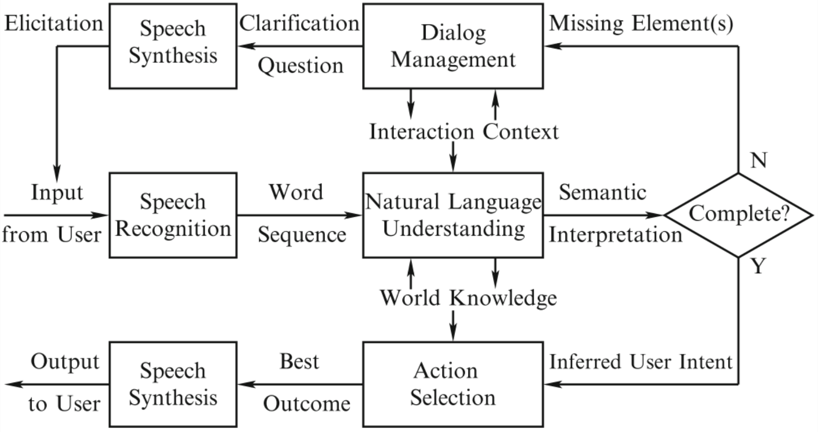
- Good speech recognition, speech synthesis and semantic interpretation
- Good interaction via dialog management
- Personalised and context aware [2]

Motivation -cont.-

“Schedule a meeting with John Monday at 2pm.”

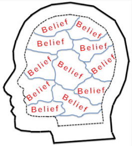
- Recognize user’s intend to create a meeting
- Act on this intend. Create calendar entry, write email to John...
- Robustness: Deal with ambiguities (which John? which Monday?...)

Intelligent Personal Assistant – Interaction Model



[3]

Semantic Interpretation



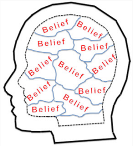
Statistical Framework

- Data driven approach on semantic interpretation POMDP:
- Maintain a system of beliefs

$\rightarrow$ Update beliefs using Bayesian inference underlying a poilicy

- Optimize policy using reinforcement learning

$\rightarrow$ Learn statistical dirstributions via observations, infer posterior

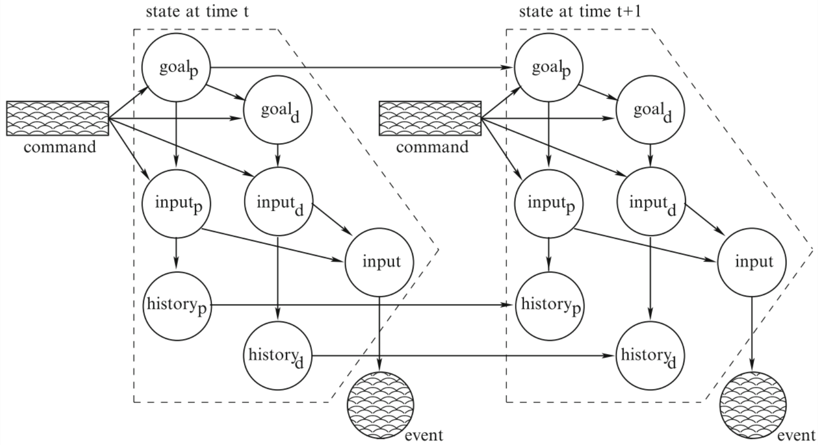


Statistical Framework - POMDPs

- Problems:
- Complex internal state: user's goal + user's input (signtificant uncertainty)+ dialogue history
- Complex mapping from dialogue states to possibly large action space

→ POMDPs usually involve a lot of approximations (rank and prune state values, invoke independence assumptions, summary state space...)

Semantic Interpretation - Statistical Framework



[3]

Rule-based Framework

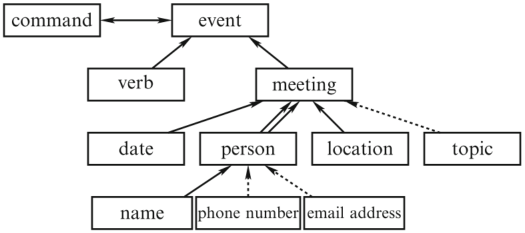
- Relies on inference enginge operating on a knowledge base
- Traditionally domain specific with a complex architecture of many components

Active Platform:

- More light-weight and developer friendly
- Loosely coupled services with specialised task representations

→ Active ontology for every task

Semantic Interpretation - Rule-based Framework



[3]

- Dynamic processing in a relational network of concepts
- Concepts allow various instantiations of canonical objects
  "monday at 2pm", "tomorrow morning"
  $\rightarrow$ date(DAY, MONTH, YEAR, HOURS, MINUTES)
- Optional, mandatory, unique and multiple children nodes

Which problems might arise when relying solely on a rule-based (active) platform?

Learning by Instruction Agent

Learning by Instruction

- Commercial systems (Google Now, Siri, Cortana...) limited to predefined commands
- Learn new commands from natural language instructions
- Learning by instruction agent (LIA) [1]
- Teach agent how to achieve commands through sequence of steps
- Lexicon Induction: Ability to generalise across taught commands:
  "Forward e-mail to Lisa." $\rightarrow$ "Forward email to Ben."

Learning by Instruction

"I'm stuck in traffic and will be late."

Instructions:

- "First, use GPRs to estimate time of arrival."
- "See who I am meeting."
- "Send an email to this person indicating that I'll be late."

$\rightarrow$ System now understands how to handle similar situations in the future.

- Also possible to teach agent the same action sequence for different commands:

  "Send an e-mail to Lisa." = "Drop Lisa an e-mail"

LIA - Learning by Instruction Agent

LIA

- Operates in an e-mail domain. Actions: Read, Compose, Send...
- Interaction via a text dialogue.
- Semantic parser (assign semantics to commands): Map commands to logic form
- Back-end (execute commands): built-in executables + declarative knowledge base

LIA - Learning by Instruction Agent

Instructing LIA

1. Teach new declarative knowledge:
   - Define new concepts along with fields and instances
   - “contact” $\rightarrow$ “has an email address”, ”Lisa is a contact”
2. Teach new procedural knowledge:
   - How to execute a new command
   - Which already known executables to use to achieve the task

Semantic Parser

- Combinatory Categorial Grammar (CCG) parser:
  Words behave like functions

1. Lexicon
   - Table mapping words to syntactic categories and logical forms
   - syntactic category: how entry can combine with other words
2. Set of grammar rules
   - Decompose phrases via function operations (application, composition...)
3. Trained parameter vector
   - Multiple parses possible, train parser to select best solution

LIA - Learning by Instruction Agent

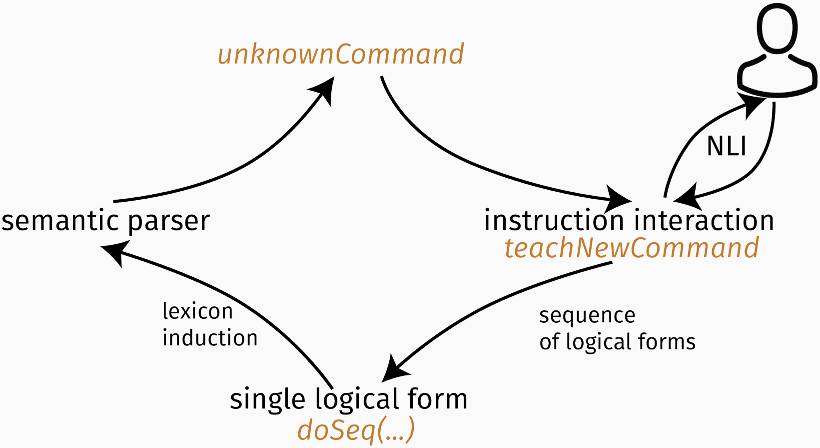
Back-end Command Executer

- Evaluate logical forms
- Primitive executable functions predefined

  sendEmail, *addFieldToConcept*, *createInstance*...
- Important functions for learning new commands:

  unknownCommand, *teachNewCommand*

Learning new Commands



Lexicon Induction

- Given the instructed command, update semantic parser to generalise
- Find subexpressions

  Spans which construct logical forms that are part of the complete logical form
- Declare subexpressions as arguments that can be filled during parsing
- “forward to Lisa” $\rightarrow$ “Lisa” parses to *lisa* $\rightarrow$ subexpression and thus argument to “forward to”

Semantic Parser training examples

Text Command	Logical Form
set the subject to time to go	(setFieldFromString (getMutableFieldByFieldName subject) (stringValue "time to go"))
send the email	(send email)
set body to email's body and send email	(doSeq (setFieldFromFieldVal (getMutableFieldByFieldName body) (evalField (getFieldByInstanceNameAndFieldName email body))) (send email))
add length as a field in table	(addFieldToConcept table (stringNoun "length"))
forward to charlie	(doSeq (doSeq (doSeq (doSeq (createInstanceByConceptName outgoingemail) (setFieldFromFieldVal (getMutableFieldByFieldName subject) (evalField (getFieldByInstanceNameAndFieldName email subject)))) (setFieldFromFieldVal (getMutableFieldByFieldName body) (evalField (getFieldByInstanceNameAndFieldName email body)))) (setFieldFromFieldVal (getMutableFieldByFieldName recipient) (evalField (getFieldByInstanceNameAndFieldName charlie email)))) (sendEmail))

[1]

Lexicon entries

Word	Syntactic Category	Logical Form
set	((S/PP_StringV)/MutableField)	(lambda x y (setFieldFromString x y))
to	PP_StringV/StringV	(lambda x x)
subject	FieldName	subject
send	S/InstanceName	(lambda x (send x))
email	InstanceName	email
set	((S/PP_FieldVal)/MutableField)	(lambda x y (setFieldFromFieldVal x y))
to	PP_FieldVal/FieldVal	(lambda x x)
and	(S/S)\S	(lambda x y (doSeq x y))
's	((Field\InstanceName)/FieldName)	(lambda x y (getFieldByInstanceNameAndFieldName y x))
forward	(S/InstanceName)	(lambda x (doSeq (doSeq (doSeq (doSeq (createInstanceByConceptName outgoingemail) (setFieldFromFieldVal (getMutableFieldByFieldName subject) (evalField (getFieldByInstanceNameAndFieldName email subject)))) (setFieldFromFieldVal (getMutableFieldByFieldName body) (evalField (getFieldByInstanceNameAndFieldName email body)))) (setFieldFromFieldVal (getMutableFieldByFieldName recipient) (evalField (getFieldByInstanceNameAndFieldName x email)))) (sendEmail)))

[1]

Conclusion

Conclusion

- The development of an intelligent personal assistant involves several complex tasks

  speech recognition, speech synthesis, dialogue management, semantic interpretation...
- Understanding the user's intent plays a crucial role
- Both statistical and rule-based semantic interpretation reveal benefits and drawbacks
- Instructable agents can help rule-based systems to improve on problems related to predefined commands and make the system more flexible

A. Azaria, J. Krishnamurthy, and T. M. Mitchell.
Instructable Intelligent Personal Agent.
2016.

W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals.
Listen, attend and spell.
arXiv preprint, pages 1–16, 2015.

F. Morbini, D. Devault, K. Sagae, J. Gerten, A. Nazarian, and D. Traum.
Natural Interaction with Robots, Knowbots and Smartphones.
Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice, pages 313–325, 2014.