Bootstrapping incremental dialogue systems from minimal data: the generalisation power of dialogue grammars

Arash Eshghi, Igor Shalyminov, Oliver Lemon
Heriot-Watt University

Presenter: Prashant Jayannavar

Problem

- Inducing task-based dialog systems
  - Example: Restaurant search

Motivation

- Poor data efficiency

- Annotation costs
    - task-specific semantic/pragmatic annotations

- Lack of support for natural spontaneous dialog/incremental dialog phenomena
    - E.g.: "I would like an LG laptop **sorry uhm** phone", "we will be **uhm** eight"

Contributions

- Solution
  - An incremental semantic parser + generator trained with RL
  - End-to-end method

- Show the following empirically:
  - Generalization power
  - Data efficiency

Background

- DS-TTR parsing (**D**ynamic **S**yntax - **T**ype **T**heory with **R**ecords)
  - Dynamic Syntax
    - word-by-word incremental and semantic grammar formalism

  - Type Theory with Records
    - Record Types (RTs): richer semantic representations

Background

- DS-TTR parsing (**D**ynamic **S**yntax - **T**ype **T**heory with **R**ecords)

$$
\begin{bmatrix} event & : e_s \\ p1_{=today(event)} & : t \end{bmatrix}
\mapsto
\begin{bmatrix} event_{=arrive} & : e_s \\ p1_{=today(event)} & : t \\ p2_{=pres(event)} & : t \\ x_{=robin} & : e \\ p3_{=subj(event,x)} & : t \end{bmatrix}
\mapsto
\begin{bmatrix} event_{=arrive} & : e_s \\ p1_{=today(event)} & : t \\ p2_{=pres(event)} & : t \\ x_{=robin} & : e \\ p3_{=subj(event,x)} & : t \\ x1 & : e \\ p3_{=from(event,x1)} & : t \end{bmatrix}
\mapsto
\begin{bmatrix} event_{=arrive} & : e_s \\ p1_{=today(event)} & : t \\ p2_{=pres(event)} & : t \\ x_{=robin} & : e \\ p_{=subj(event,x)} & : t \\ x1_{=Sweden} & : e \\ p3_{=from(event,x1)} & : t \end{bmatrix}
$$

"A: Today" $\mapsto$ *"..Robin arrives"* $\mapsto$ *"B: from?"* $\mapsto$ *"A: Sweden"*

BABBLE

- Treat natural language generation (NLG) and dialog management (DM) as a joint decision problem
  - Given a “dialog state” decide what to say

  - Learn to do this through learning a policy (π : S -> A) -- RL

  - Define “dialog state” using output of the DS-TTR parser

BABBLE

- Inputs:
  - A DS-TTR parser
  - A dataset D of dialogs in target domain

- Output:
  - Policy π : S -> A (given a “dialog state” deciding what to say)

BABBLE

- MDP setup
    - S: set of all dialog states (induced from dataset D)
    - A: set of all actions (words in the DS lexicon)
    - G_d: Goal state
    - R: reaching G_d while minimizing dialog length

BABBLE

- Dialog state:
    - **Between SYSTEM and USER utterances and between every word of SYSTEM utterances**

BABBLE

- Dialog state:
  - **Between SYSTEM and USER utterances and between every word of SYSTEM utterances**

SYSTEM: **[S_0]** What **[S_1]** would **[S_2]** you **[S_3]** like **[S_4]** ? **[S_5 = S_trig_1]**

USER: A phone **[S_6]**

SYSTEM: by **[S_7]** which **[S_8]** brand **[S_9]** ? **[S_10 = S_trig_2]**

USER: …

BABBLE

- Dialog state:
  - Between SYS and USER utterances and between every word of SYS utterances
  - **Context up until that point in time**
  - **Context C = <c_p, c_g>**

BABBLE

- SYSTEM: What would you like ?
  USER: A phone
  SYSTEM: by which brand ? [S_10]

BABBLE

Grounded Semantics

Sys: What would you like?

Usr: a phone

$$\left[\begin{array}{lcl} x2 & : & e \\ e2_{=like} & : & es \\ x1_{=USR} & : & e \\ p2_{=pres(e2)} & : & t \\ p5_{=subj(e2,x1)} & : & t \\ p4_{=obj(e2,x2)} & : & t \\ p11_{=phone(x2)} & : & t \end{array}\right]$$

Current Turn Semantics

Sys: by which brand?

$$\left[\begin{array}{lcl} x2 & : & e \\ e2_{=like} & : & es \\ x1_{=USR} & : & e \\ p2_{=pres(e2)} & : & t \\ p5_{=subj(e2,x1)} & : & t \\ p4_{=obj(e2,x2)} & : & t \\ p11_{=phone(x2)} & : & t \\ x3 & : & e \\ p10_{=by(x2,x3)} & : & t \\ p9_{=brand(x3)} & : & t \\ p10_{=question(x3)} & : & t \end{array}\right]$$

BABBLE

- Dialog state:
  - Between SYS and USER utterances and between every word of SYS utterances
  - Context up until that point in time
  - Context C = <c_p, c_g>
  - **State encoding function F: C -> S maps context to a binary vector**

Grounded Semantics

$$\begin{bmatrix} x2 & : & e \\ e2_{=like} & : & es \\ x1_{=USR} & : & e \\ p2_{=pres(e2)} & : & t \\ p5_{=subj(e2,x1)} & : & t \\ p4_{=obj(e2,x2)} & : & t \\ p11_{=phone(x2)} & : & t \end{bmatrix}$$

Current Turn Semantics

$$\begin{bmatrix} x2 & : & e \\ e2_{=like} & : & es \\ x1_{=USR} & : & e \\ p2_{=pres(e2)} & : & t \\ p5_{=subj(e2,x1)} & : & t \\ p4_{=obj(e2,x2)} & : & t \\ p11_{=phone(x2)} & : & t \\ x3 & : & e \\ p10_{=by(x2,x3)} & : & t \\ p9_{=brand(x3)} & : & t \\ p10_{=question(x3)} & : & t \end{bmatrix}$$

Dialogue so far

SYS: What would you like?
USR: a phone
SYS: by which brand?

RT Feature: $\begin{bmatrix} x10 & : & e \\ p15_{=brand(x10)} & : & t \end{bmatrix} \begin{bmatrix} e3_{=like} & : & es \\ p2_{=pres(e3)} & : & t \end{bmatrix} \begin{bmatrix} x10 & : & e \\ x8 & : & e \\ p14_{=by(x8,x10)} & : & t \end{bmatrix} \begin{bmatrix} e3_{=like} & : & es \\ x5_{=usr} & : & e \\ p7_{=subj(e3,x5)} & : & t \end{bmatrix} \begin{bmatrix} x8 & : & e \\ e3_{=like} & : & es \\ p6_{=obj(e3,x8)} & : & t \end{bmatrix}$

$$\text{State: } \left\langle \begin{array}{lccccc} & F_1 \downarrow & F_2 \downarrow & F_3 \downarrow & F_4 \downarrow & F_5 \downarrow \\ \text{Current Turn:} & 1, & 1, & 1, & 1, & 1, \\ \text{Grounded:} & 0, & 1, & 0, & 1, & 1 \end{array} \right\rangle$$

BABBLE

- Dialog state:
    - Between SYS and USER utterances and between every word of SYS utterances
    - Context up until that point in time
    - Context C = <c_p, c_g>
    - State encoding function F: C -> S maps context to a binary vector

BABBLE

RL to solve the MDP

SYSTEM: [S_0] What [S_1] would [S_2] you [S_3] like [S_4] ? [S_5 = S_trig_1]

USER: A phone [S_6] **<- Simulated User**

SYSTEM: by [S_7] which [S_8] brand [S_9] ? [S_10 = S_trig_2]

USER: … **<- Simulated User**

SYSTEM: …

BABBLE

User simulation

- Generate user turns based on context
- Monitor system utterance word-by-word

BABBLE

User simulation

- **Generate user turns based on context**
  - Run parser on dataset D and extract rules of the form:

S_trig_i -> {u_1, u_2, …, u_n}

S_trig_i = a trigger state

u_i = user utterance following S_trig_i in D

BABBLE

SYSTEM: [S_0] What [S_1] would [S_2] you [S_3] like [S_4] ? **[S_5 = S_trig_1]**

USER: A phone [S_6] **<- Simulated User**

SYSTEM: by [S_7] which [S_8] brand [S_9] ? **[S_10 = S_trig_2]**

USER: … **<- Simulated User**

SYSTEM: …

BABBLE

User simulation

- Generate user turns based on context
- **Monitor system utterance word-by-word**
    - After system generates a word, check if new state **subsumes** one of the S_trig_i
    - If not, penalize system and terminate learning episode

BABBLE

SYSTEM: [S_0] What **[S_1]** would **[S_2]** you **[S_3]** like [S_4] ?

USER: A phone [S_6]

SYSTEM: by [S_7] which [S_8] brand [S_9] ?

USER: …

SYSTEM: …

Evaluation

- 2 datasets to test generalization:
    - bAbI
        - Dataset of dialogs by Facebook AI Research
        - Goal oriented dialogs for restaurant search
        - API call at the end

Evaluation

- bAbI+
    - Add incremental dialog phenomena to bAbI

    - Hesitations: "we will be **uhm** eight"
    - Corrections: "I would like an LG laptop **sorry uhm** phone"

    - These phenomena mixed in probabilistically
        - Affect 11336 utterances in the 3998 dialogs

Evaluation

- **Approach to compare to (MEMN2N):**
  - Bordes and Weston 2017: Learning end-to-end goal-oriented dialog
  - Uses memory networks
  - Retrieval based model

Evaluation

- **Experiment 1: Generalization from small data**
  - Do not use the original system for a direct comparison
    - Use a retrieval based variant

  - 1-5 examples from bAbI train set
  - Test on 1000 examples from bAbI test set
  - Test on 1000 examples from bAbI+ test set

Evaluation

- **Experiment 1: Generalization from small data**
  - **Metric: Per utterance accuracy**

# of training dialogues:	1	2	3	4	5
BABBLE on bAbI	67.12	73.36	72.63	73.32	74.08
MEMN2N on bAbI	2.77	59.15	70.94	71.68	72.6
BABBLE on bAbI+	59.42	65.27	63.45	64.34	65.2
MEMN2N on bAbI+	0.22	56.75	68.65	71.84	73.2

Evaluation

- **Experiment 2: Semantic Accuracy**
    - **Metric: Accuracy of API call**
    - BABBLE: 100% on both bAbI and bAbI+

    - MEMN2N: Nearly 0 on both bAbI and bAbI+
    - MEMN2N (when trained on full bAbI dataset): 100% on bAbI and only 28% on bAbI+

Summary

- An incremental semantic parser + generator trained with RL
- End-to-end training
- Support incremental dialog phenomena
- Showed the following empirically:
  - Generalization power
  - Data efficiency