Task-Oriented Dialogue as Dataflow Synthesis

Jacob AndreasJohn BufeDavid BurkettCharles ChenJosh ClausmanJean CrawfordKate CrimJordan DeLoachLeah DornerJason EisnerHao FangAlan GuoDavid HallKristin HayesKellie HillDiana HoWendy IwaszukSmriti JhaDan KleinJayant KrishnamurthyTheo LanmanPercy LiangChristopher H. LinIlya LintsbakhAndy McGovernAleksandr NisnevichAdam PaulsDmitrij PettersBrent ReadDan RothSubhro RoyJesse RusakBeth ShortDiv SlominBen SnyderStephon StriplinYu SuZachary TellmanSam ThomsonAndrei VorobevIzabela WitoszkoJason WolfeAbby WrayYuchen ZhangAlexander Zotov

Microsoft Semantic Machines <sminfo@microsoft.com>

Motivation

- New approach to task-oriented dialogue
- So far, there are two broad ways of approaching this problem
 - Fixed symbolic state representations (much like state machines, slotfilling systems)
 - Deep continuous state representations (like using DL models)
- The proposed framework picks up the strength of both approaches by having new
 - state representations (data flow graphs)
 - agent architecture that interact and extend these state representation

Dataflow graphs

Executable description of the entities that were mentioned and the actions and computations that produced them

User: Can you create a meeting with Megan right before that starts?

```
User: Where is my meeting at 2 this afternoon?
place(findEvent(EventSpec(start=pm(2))))
```



Agent: It's in Conference Room D.

```
createEvent(EventSpec(
    end=start(<u>refer(Constraint[Event]())),
    attendee=PersonSpec(name='Megan')</u>
```

))



Agent: Which person named Megan did you mean?

User: Megan Bowen.

revise(
 new=PersonSpec(name='Megan Bowen'),
 oldLoc=Constraint[PersonSpec](),
 rootLoc=RoleConstraint(output)



Agent: *Ok, I'll book a meeting with Megan Bowen from 1:30 to 2:00 pm today.*

Overview of the framework – graph and evaluation



Overview of the framework – Reference



Overview of the framework – Revision



Reference resolution

Any introduced entities can be referred to again. This referral may or may not be overt:

> Agent: You have lunch at 1p User: What is after that? Agent: You have a meeting. User: what time?

...

...

Reference resolution

When is Easter?



Schedule a meeting day after that.







Reference resolution - constraints

Used to constrain the returned node if that information is available.

- Type constraint
- Property constraint
- Role constraint

User: *What's happening this morning?* findEvent(EventSpec(start=and(today(), during(morning())))) You have a planning meeting at 9 am. Agent: What do I have after that? User: findEvent(EventSpec(start=after(end(refer(Constraint[Event]())))) Your birthday lunch is at 12:30 pm. Agent: User: How many people are coming to the 9 am meeting? length(attendees(refer(Constraint[Event](start=am(9)))) Agent: 5 people will be at the planning meeting.

Salience retrieval model

- The framework is agnostic to the implementation of this model.
- For the paper a hard-coded heuristic is used where root node of previous utterance is used if the constraints are satisfied.
- If a satisfying node is not found, other recent utterances are checked.
- If nothing is found still, a query is sent to a database:

Revision

These happen when there are updates to previously asked queries. Previously constructed sub graphs can directly be used this way.

- User: What time on Tuesday is my planning meeting?
- start(findEvent(EventSpec(
 name='planning',
 - start=DateTimeSpec(weekday=tuesday))))
- Agent: You meet with Grace at noon.
- User: Sorry, I meant all-hands.
- Agent: Your all-hands meeting is at 2:30 pm.

Revision

User: What time on Tuesday is my planning meeting?

start(findEvent(EventSpec(

name='planning',

start=DateTimeSpec(weekday=tuesday))))

Agent: You meet with Grace at noon.

User: Sorry, I meant all-hands.

Agent: Your all-hands meeting is at 2:30 pm.

User: Sorry, I meant all-hands.

revise(rootLoc=RoleConstraint(output),
 oldLoc=Constraint[String](),
 new='all-hands')

Arguments to **revise()**:

- rootLoc, a constraint to find the top-level node of the original computation
- oldLoc, a constraint on the node to replace within the original computation
- new, a new graph fragment to substitute there.

Revision

User: What time on Tuesday is my planning meeting?

start(findEvent(EventSpec(
 name='planning',
 start=DateTimeSpec(weekday=tuesday))))

- Agent: You meet with Grace at noon.
- User: Sorry, I meant all-hands.
- Agent: Your all-hands meeting is at 2:30 pm.

```
revise(rootLoc=RoleConstraint(output),
    oldLoc=Constraint[String](),
    new='all-hands')
```





Revision – case of missing arguments

User: *When is lunch?*

start(findEvent(EventSpec(name='lunch')))

Agent: It's at 1 pm.

User: *What about tomorrow?*

<u>revise</u>(

rootLoc=RoleConstraint(output),
oldLoc=Constraint[DateTimeSpec],
new=tomorrow())



EventSpec

findEvent

start

'lunch'

Recovery (handling exceptions)

User: *Book a meeting for me.*Agent: *When should the meeting start?*

- User: *Who is coming to the planning meeting?*
- Agent: Susan Chu and Susan Brown.
- User: What is Susan's email?

User: When is my first meeting on February 30?

Recovery (handling exceptions)

User: When is my first meeting on February 30?



Agent: There is no 30th of February. Did you mean some other date?

User: I meant February 28.	User: I meant March.
<pre>revise(rootLoc=RoleConstraint(output),</pre>	<pre>revise(rootLoc=RoleConstraint(output),</pre>
oldLoc=Constraint[DateTimeSpec](),	oldLoc=Constraint[Month],
new=DateTimeSpec(month=feb, day=28))	new=mar)

The authors provide a new richly annotated dataset SMCalFlow.

(<u>https://microsoft.github.io/task oriented dialogue as dataflow syn</u> <u>thesis/</u>)

They also produce a version of MultiWOZ dataset with dataflow-based annotations.

Experiments

	Full		Ref. Turns		Rev. Turns	
	dev	test	dev	test	dev	test
# of Turns	13,499	21,224	3,554	8,965	1,052	3,315
Dataflow inline	.729 .696	.665 .606	.642 .533	.574 .465	.697 .631	.565 .474

Joint Goal	Dialogue	Prefix
.467	.220	3.07
.447	.202	2.97
.467	.205	2.90
.454	.168	2.73
	Joint Goal .467 .447 .467 .454	Joint GoalDialogue.467.220.447.202.467.205.454.168