Goal-Oriented Chatbot Dialog Management Bootstrapping with Transfer Learning

Vladimir Ilievski[1], Claudiu Musat[2], Andreea Hossmann[2], Michael Baeriswyl[2]

[1]IC School, EPFL, Switzerland   [2]Artificial Intelligence Group, Swisscom

Goal-Oriented (GO) Chatbots

Three key elements:
1. Goal: help users achieve a predefined goal
2. Domain: e.g. movie booking
3. Slots and intents:
   inform(date = 'tomorrow')

Low in-domain data availability

- Non-trivial data requirements
- Limited in-domain data
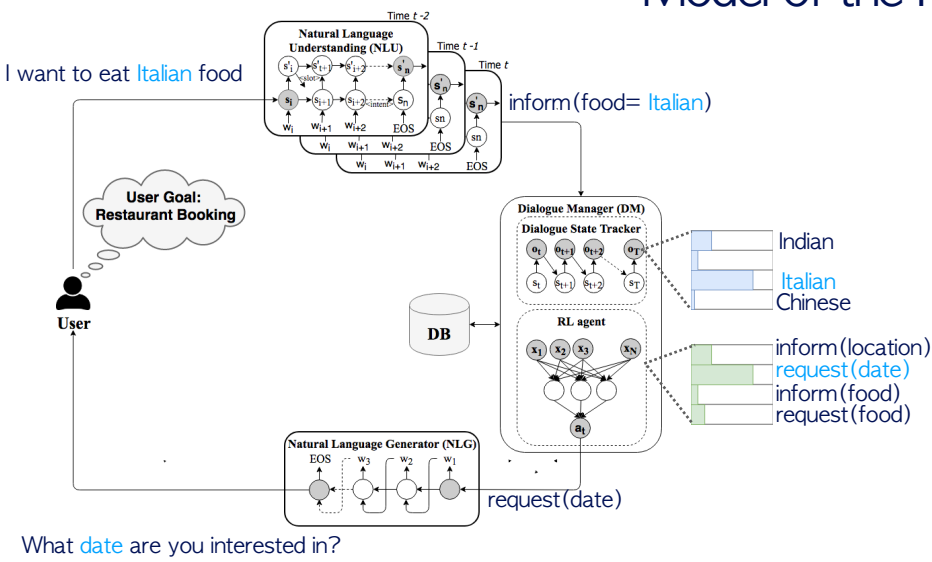- Obtaining and labeling in-domain data is hard

Leverage domain similarity

Transfer the knowledge from one SOURCE domain to another TARGET domain
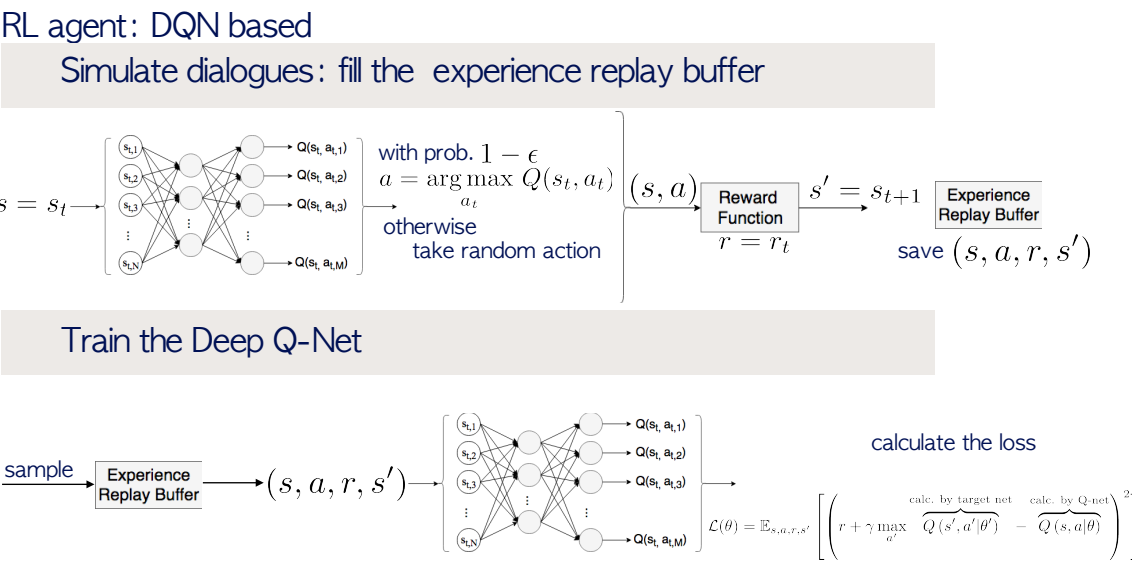
Contributions

1. Training GO chatbots with less data: models trained with transfer learning achieve better performance
2. Better GO chatbots: positive effect when all domain data is available
3. Transfer learning is complementary to the warm-starting technique

Model of the RL-based GO Chatbots

I want to eat Italian food

inform(food= Italian)

What date are you interested in?

request(date)

Modeled as a Partially Observable Markov Decision Process

RL agent: DQN based

Simulate dialogues: fill the experience replay buffer

$$s = s_t$$

with prob. $1 - \epsilon$
$a = \arg\max_{a_t} Q(s_t, a_t)$
otherwise take random action

(s, a) → Reward Function → $s' = s_{t+1}$, $r = r_t$ → Experience Replay Buffer

save (s, a, r, s')

Train the Deep Q-Net

sample → Experience Replay Buffer → (s, a, r, s')

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} \overline{Q(s', a'|\theta')} - \overline{Q(s, a|\theta)}\right)^2\right]$$
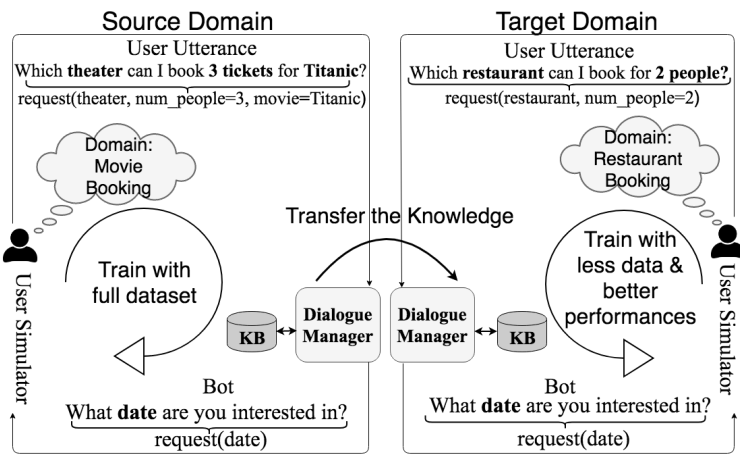
Intuition

Dialog state depends on the type of slots

Domain 1 Movie Booking		Domain 2 Restaurant Booking	
I	O	O	Where
want	O	O	in
to	O	B-City	London
book	O	O	I
tickets	O	O	can
for	O	O	find
Titanic	B-Movie	O	an
for	O	B-Food	Indian
today	B-Date	O	restaurant
somewhere	O	O	for
in	O	B-Date	tomorrow
London	B-City	I-Date	night

Share common information => Transfer the knowledge

Transfer Learning

Source Domain
User Utterance
Which **theater** can I book **3 tickets** for **Titanic**?
request(theater, num_people=3, movie=Titanic)

Domain: Movie Booking

Train with full dataset

Transfer the Knowledge

Target Domain
User Utterance
Which **restaurant** can I book for **2 people**?
request(restaurant, num_people=2)

Domain: Restaurant Booking

Train with less data & better performances

KB → Dialogue Manager ⟷ Dialogue Manager ← KB

Bot
What date are you interested in?
request(date)

Bot
What date are you interested in?
request(date)
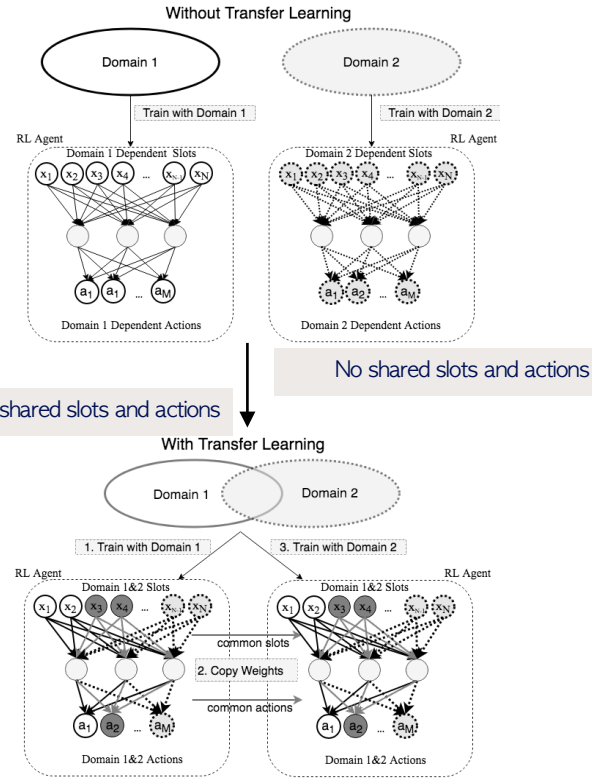
Relaxed model: no NLU and NLG units => semantic frame level execution

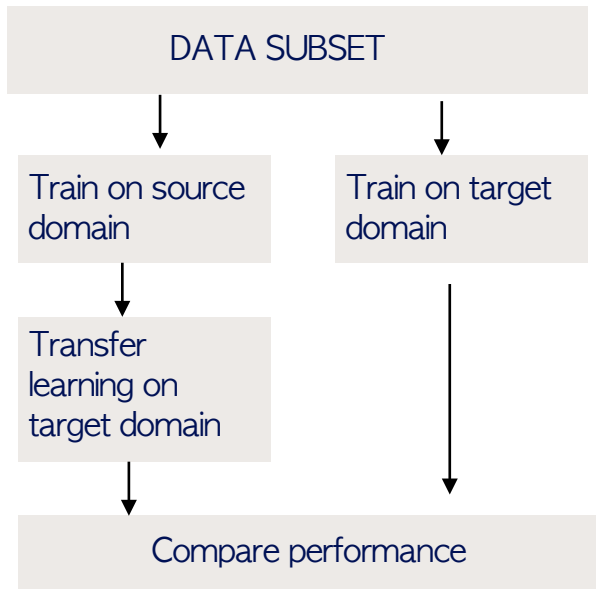Semantic Frames: set of slot-value pairs

How?

Without Transfer Learning

Domain 1 — Train with Domain 1 — RL Agent — Domain 1 Dependent Slots — Domain 1 Dependent Actions

Domain 2 — Train with Domain 2 — RL Agent — Domain 2 Dependent Slots — Domain 2 Dependent Actions

No shared slots and actions

shared slots and actions

With Transfer Learning

1. Train with Domain 1 — RL Agent — Domain 1&2 Slots — common slots — Domain 1&2 Actions

3. Train with Domain 2 — RL Agent — Domain 1&2 Slots — Domain 1&2 Actions

2. Copy Weights — common actions

How do we experiment?

Domain overlapping

SOURCE: Movie booking
TARGET: Restaurant booking

Domain extension

SOURCE: Restaurant booking
TARGET: Tourist info

DATA SUBSET

Train on source domain

Train on target domain

Transfer learning on target domain

Compare performance

Results

Total of 120 training user goals and 32 testing user goals

Subset of n user goals: (5, 10, 20, 30, 50, 120)

Train with less data

Train faster