Goal-Oriented Chatbot Dialog Management Bootstrapping with Transfer Learning

Vladimir Ilievski, Claudiu Musat,
Andreea Hossmann, Michael Baeriswyl

IJCAI, Stockholm, July 2018

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

swisscom

Agenda

1.Key elements of Goal-Oriented Chatbots

2. Problem statement and transfer learning solution

3. Model

4. Transfer Learning

5. Experiments and Results

Key Elements of the Goal-Oriented (GO) Chatbots

Domain

Predefined domain of expertise:

- movie booking
- restaurant booking

Slots And Intents

User intent:

- inform
- request

Slots or intent parameters:

- date: *tomorrow*
- count: *2 people*

Predefined Goal

Remembering user's choices.

Driving the conversation with towards achieving the goal.

Paradigms of implementations

Fully-Supervised

Sequence-to-Sequence Fashion

Encode a user request and its context

Decode a bot answer

Mimicking an expert

Require huge amounts of data

Reinforcement Learning

Based on Deep Q-Nets (DQN)

Simulate conversation

Explore the dialogue space

Limited number of dialogue turns

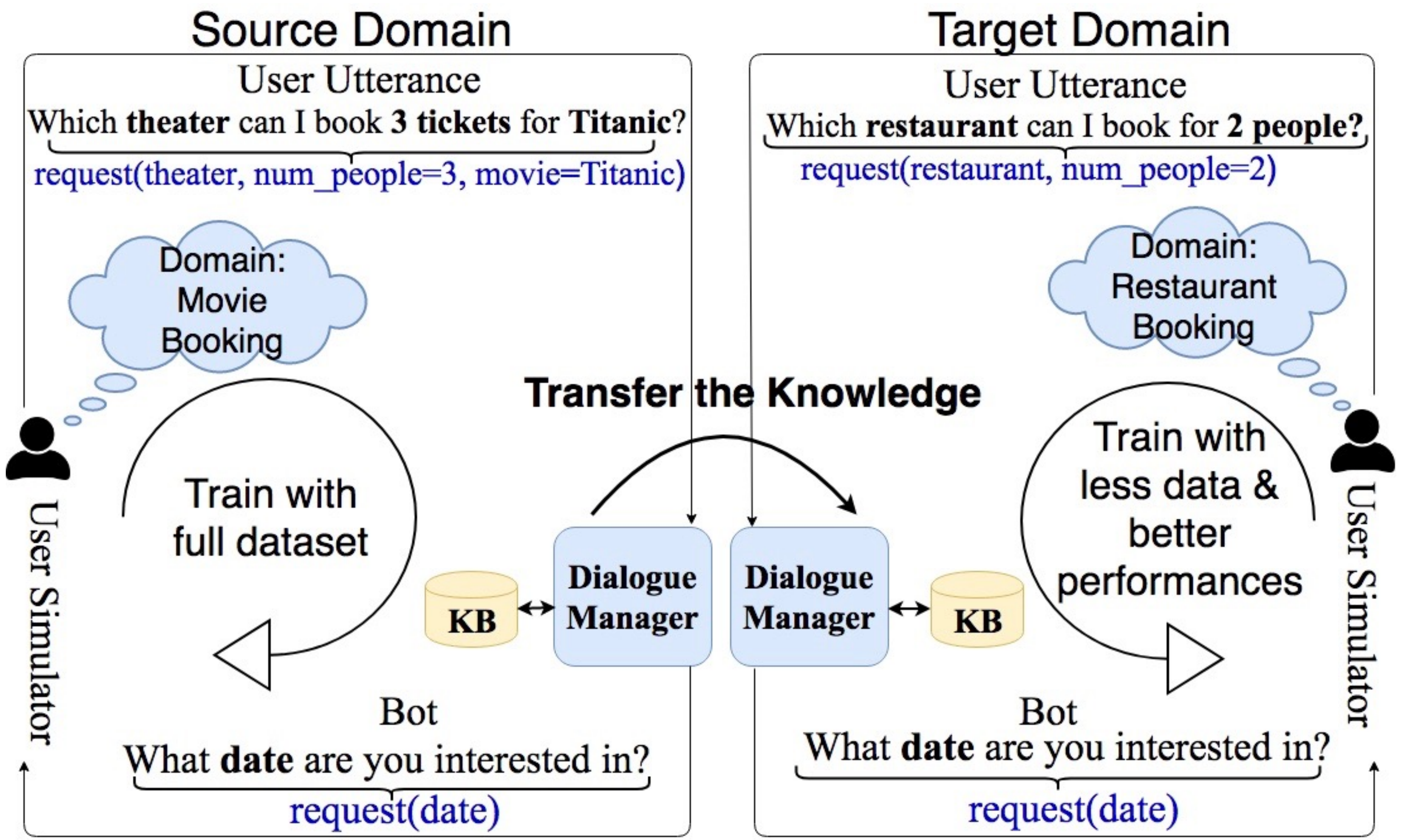
Require less data

Our choice

Problem: Limited Data

Challenge	Solution
Non trivial data requirements	Leverage the domain similarity
Limited in-domain data	Use *Transfer Learning*
Obtaining in-domain data is hard	Use less data

Solution: Transfer Learning



Goal-Oriented Dialog

- At time t:
  - given the user utterance u_t
  - the system replies with action a_t
- User utterance:
  - user's intent (e.g. inform, request info)
  - intent parameters or slots (e.g. date: *today*)
- System action:
  - Request a value for empty slot
  - Suggest a value based on a Knowledge Base

Goal-Oriented Dialog

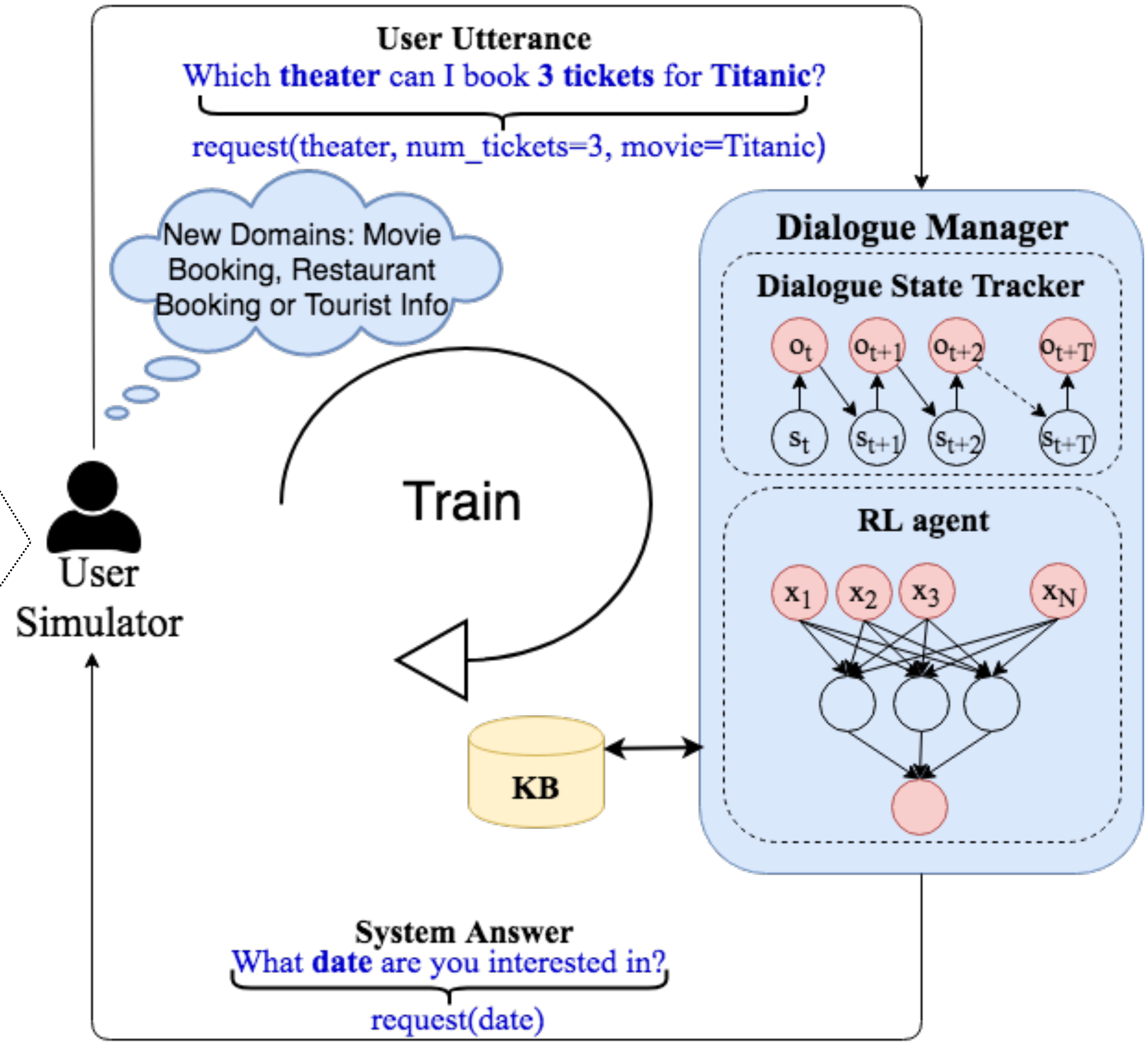
- The entire dialog: slot-value pairs called *semantic frames*
- Two levels of execution:
  - Semantic level
  - Natural language level

Model

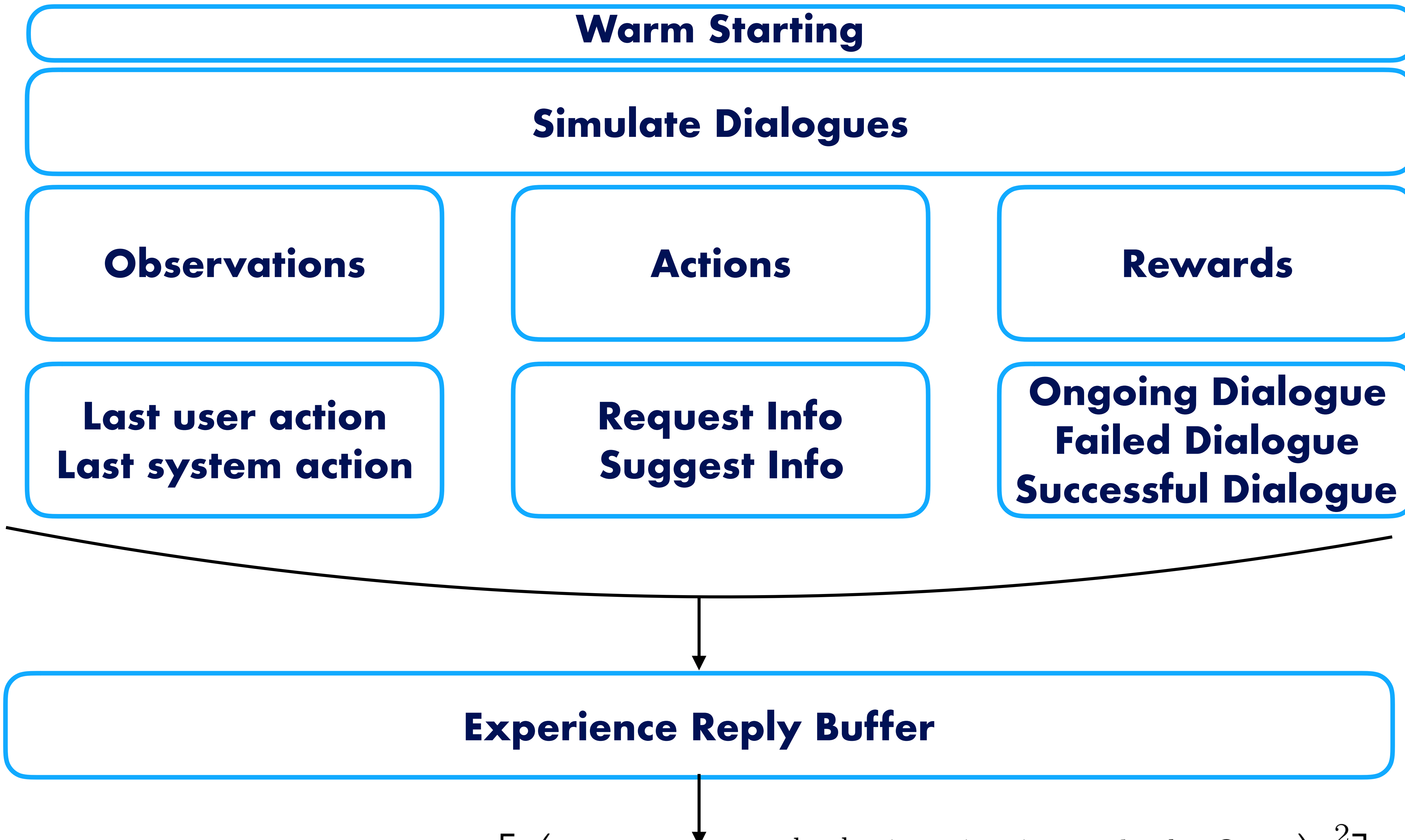
User Goal

inform_slots:
{
movie_name: "Titanic",
number_of_people: "3",
date: "tomorrow"
},

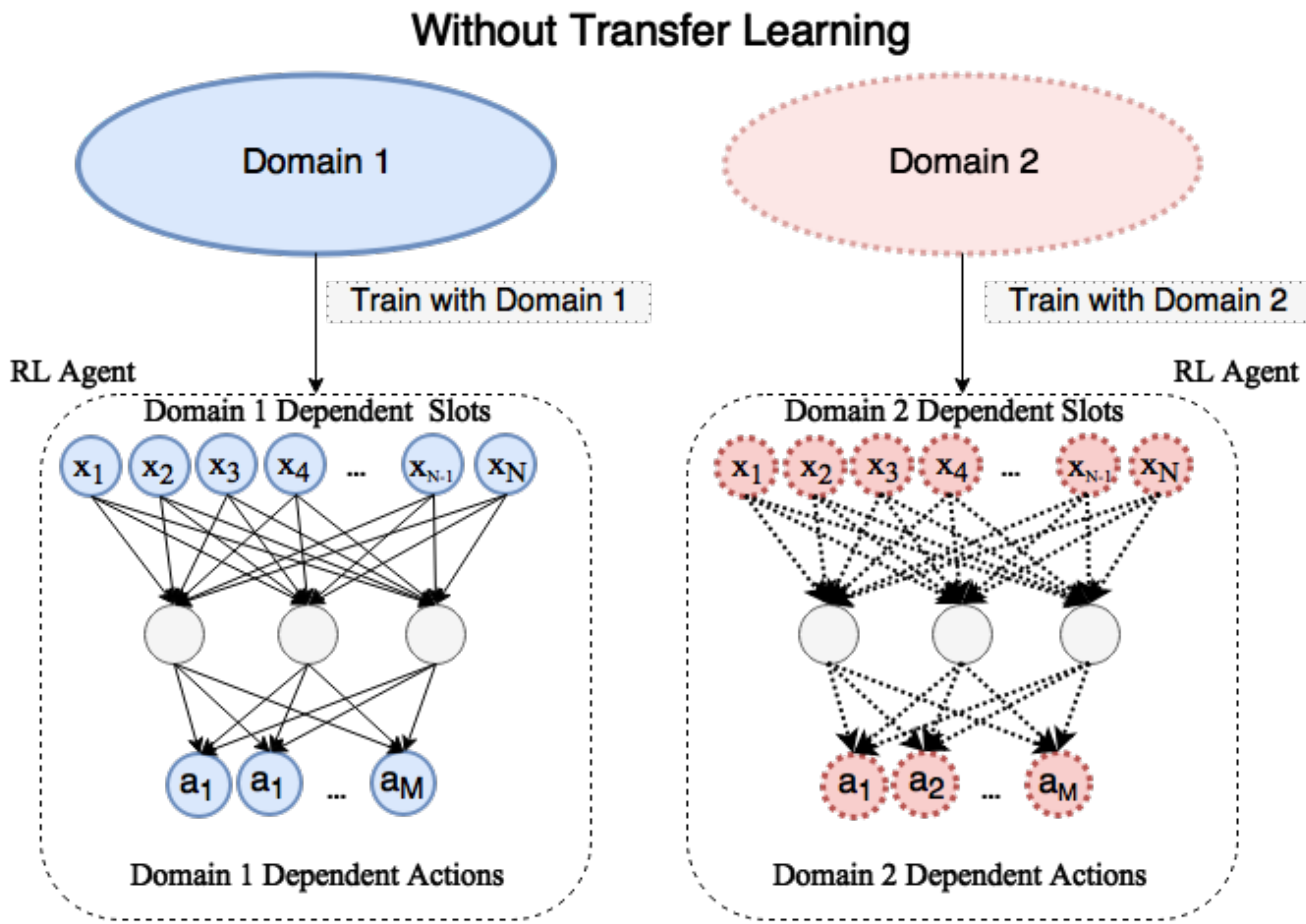
request_slots:
{
city,
theater,
start_time
}



RL Agent

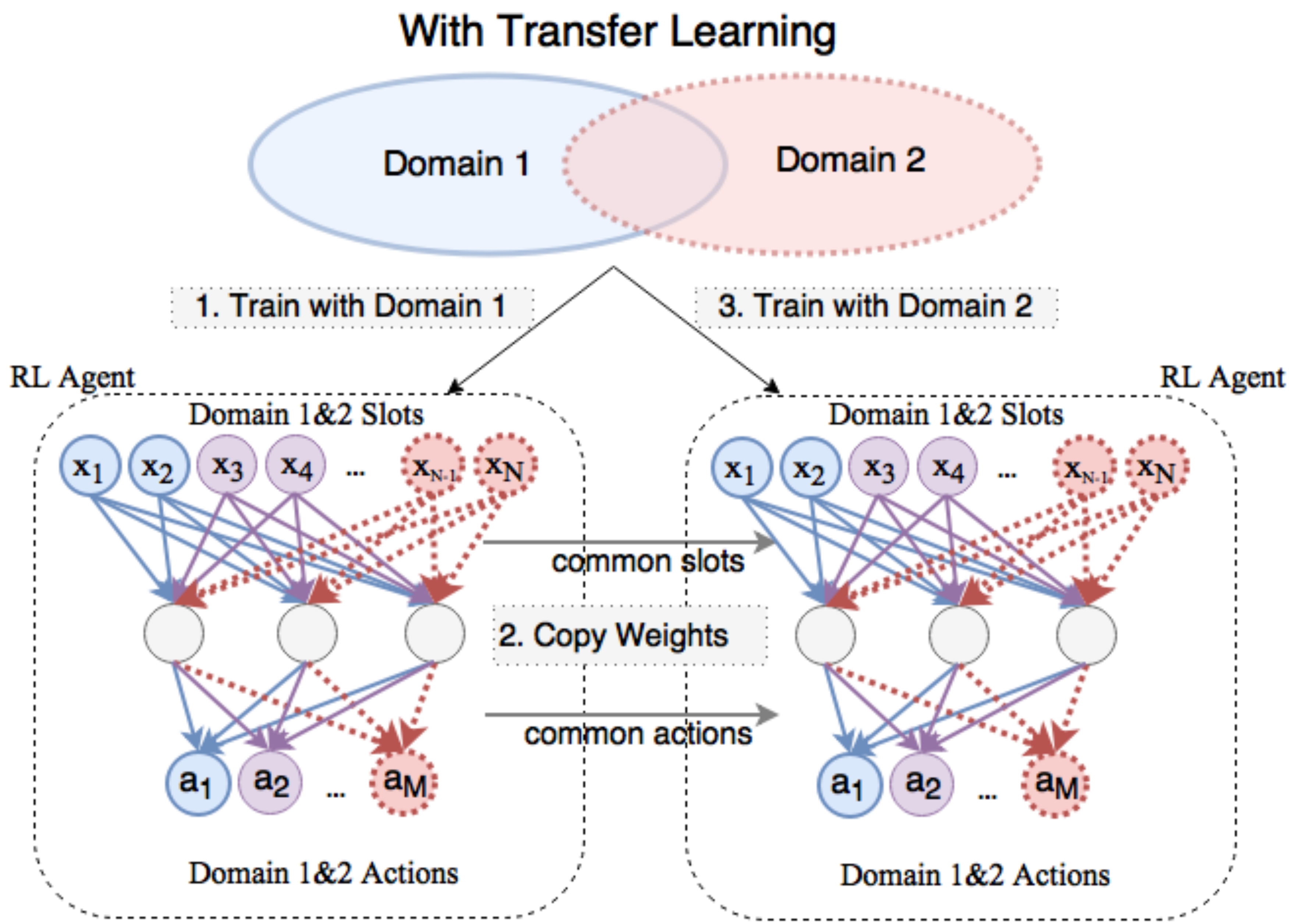


$$\mathcal{L}(\theta) = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}}\left[\left(r_t + \gamma \max_{a_{t+1}} \overbrace{Q\left(s_{t+1}, a_{t+1}|\theta'\right)}^{\text{calc. by target net}} - \overbrace{Q\left(s_t, a_t|\theta\right)}^{\text{calc. by Q-net}}\right)^2\right]$$



No Transfer Learning

No shared slots and actions - no shared weights



With Transfer Learning

Shared slots and actions - shared weights

Experiments

Data

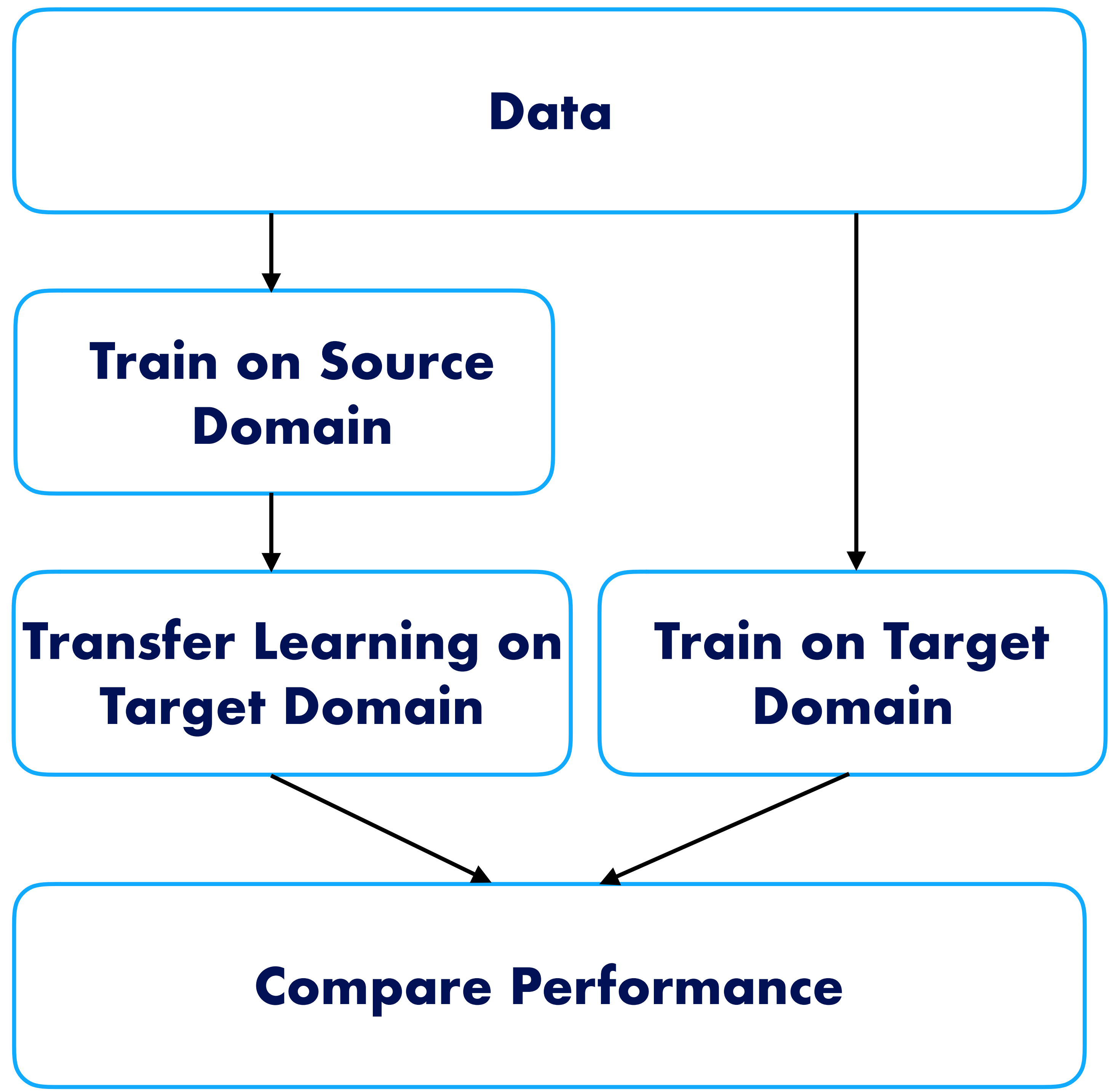
Two hypothesis:
- train with less data - compare success rate
- train faster - compare learning rate

Pair of Domains:
- Source Domain
- Target Domain

For each domain:
- 120 training user goals
- 32 testing user goals

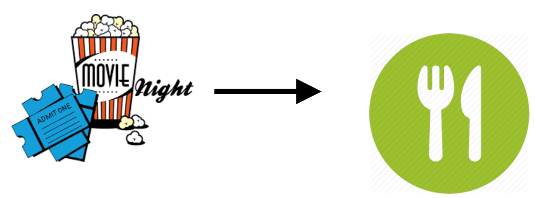
Two models:
- transfer learning model
- model from scratch

Flow

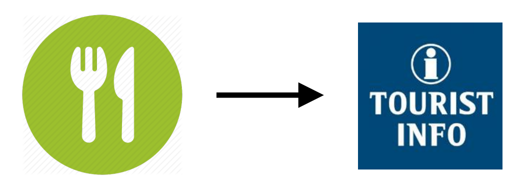


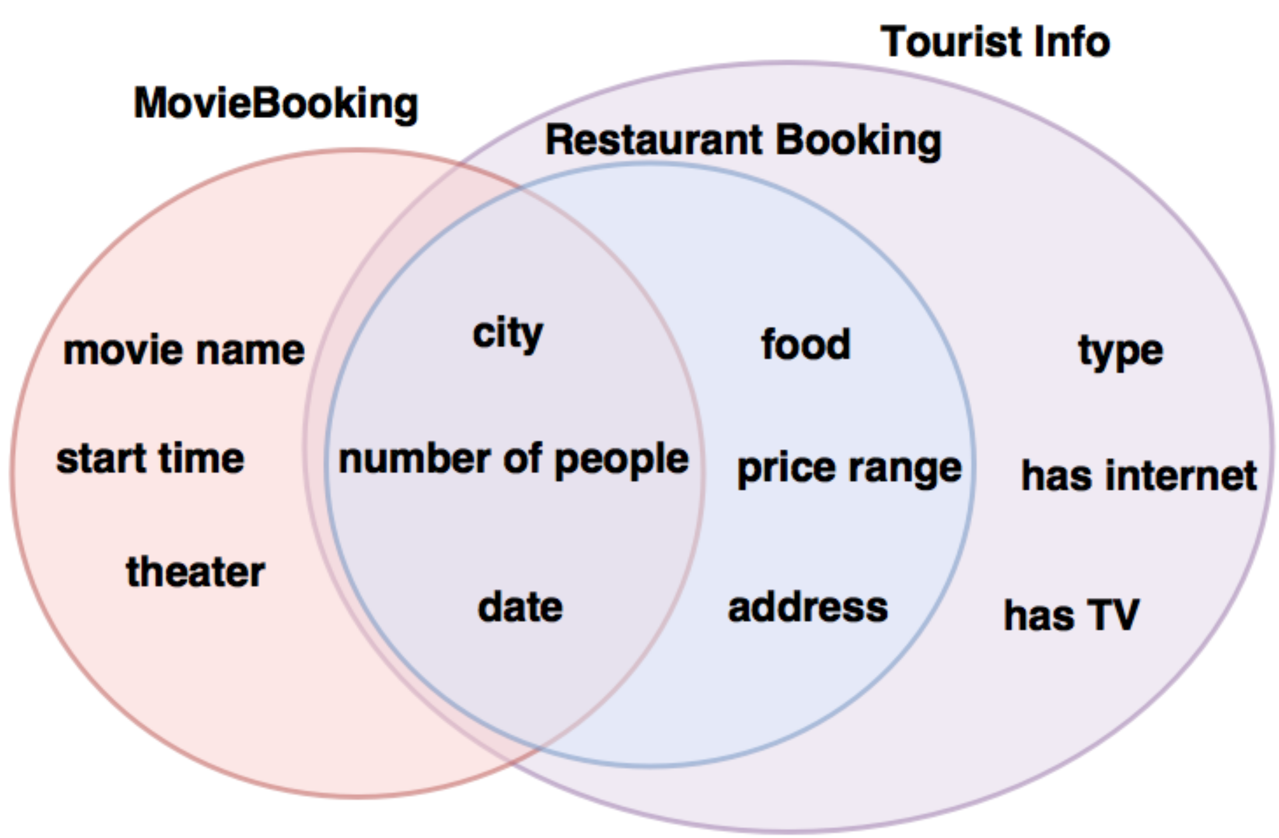
Domain Cases

1. Domain Overlapping:
- **Source Domain: Movie Booking**
- **Target Domain: Restaurant Booking**



2. Domain Extension:
- **Source Domain: Restaurant Booking**
- **Target Domain: Tourist Info**



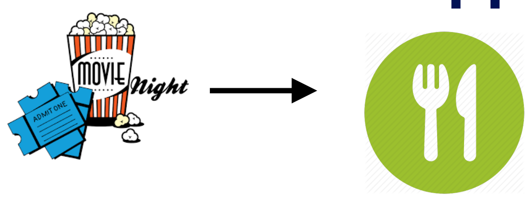


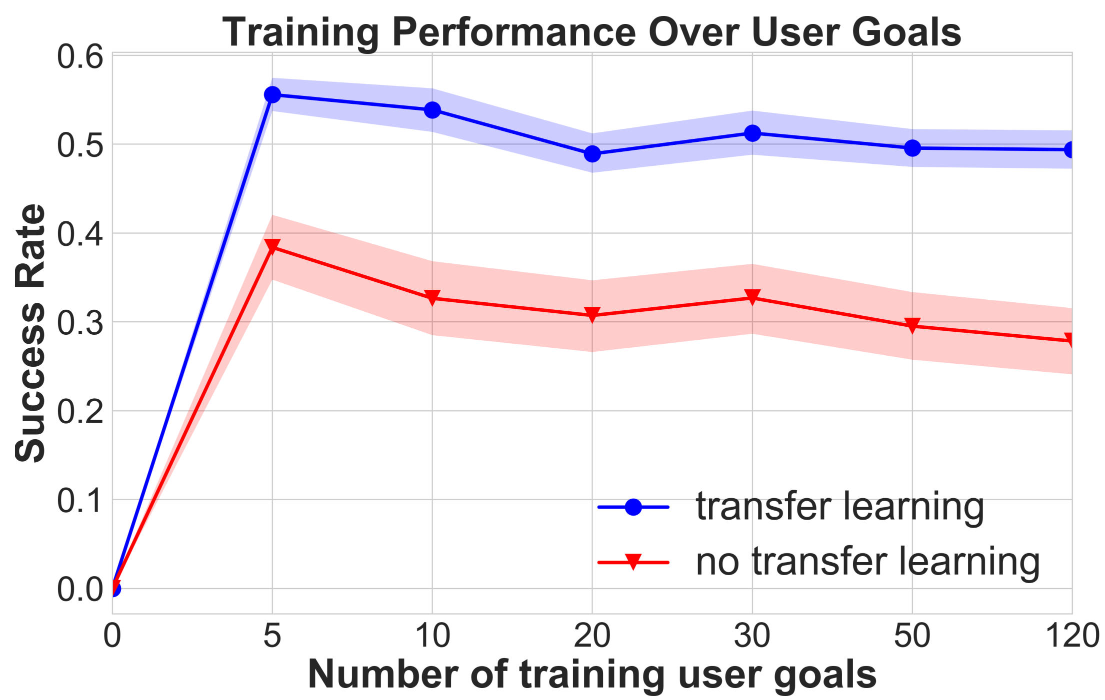
Train With Less Data

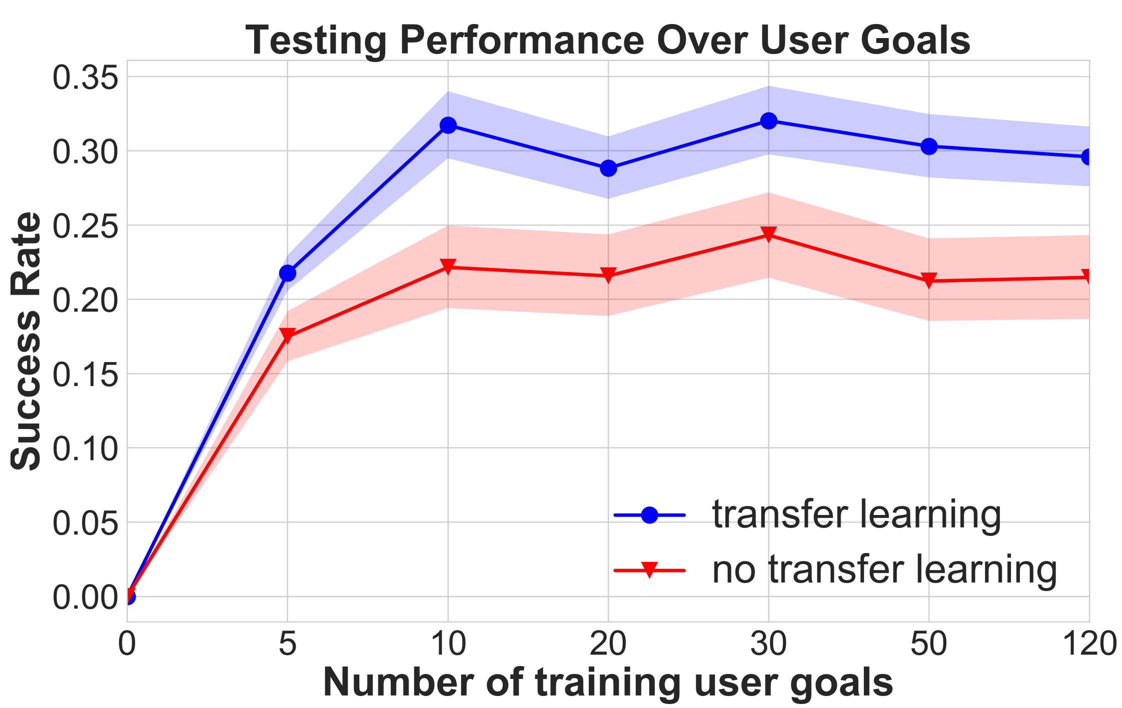
- For both models we do 100 iterations of:
  - Splitting the data set in portions: 5, 10, 20, 30, 50 and 120
  - Warm-start both models
  - Train on each subset and test on the test set of 32 user goals
  - Report the training and testing success rates

Train With Less Data - Results

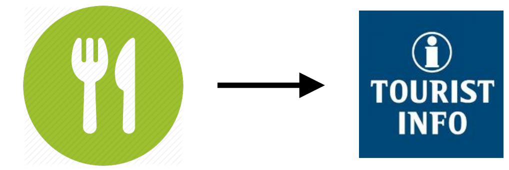
Domain Overlapping

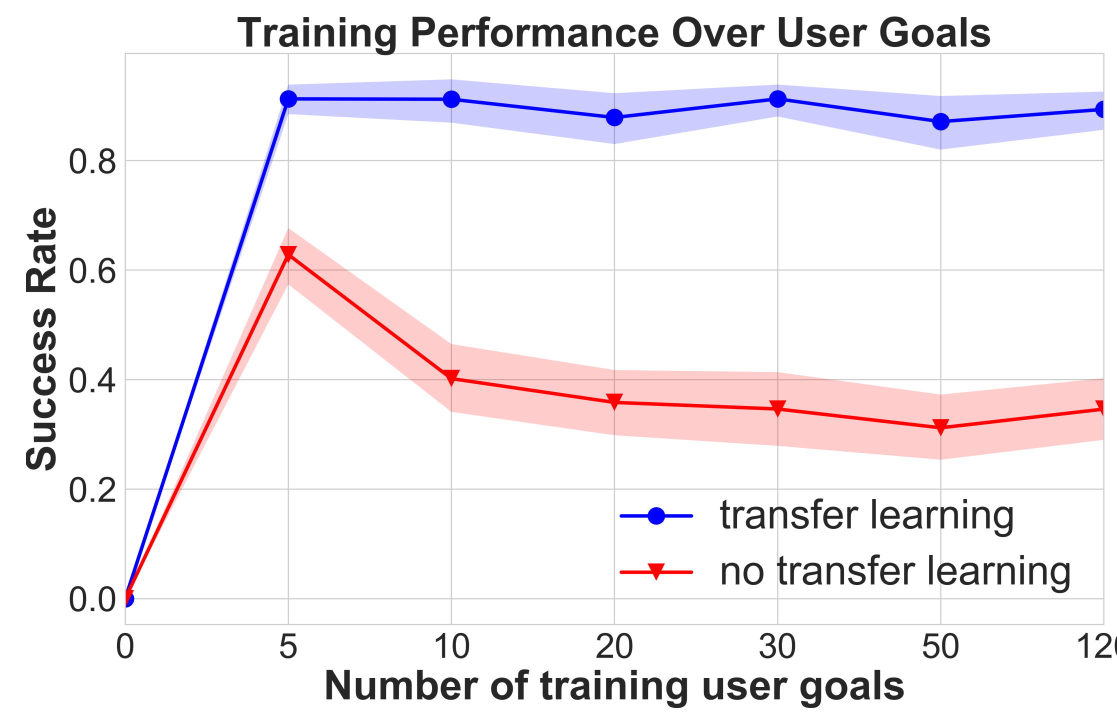


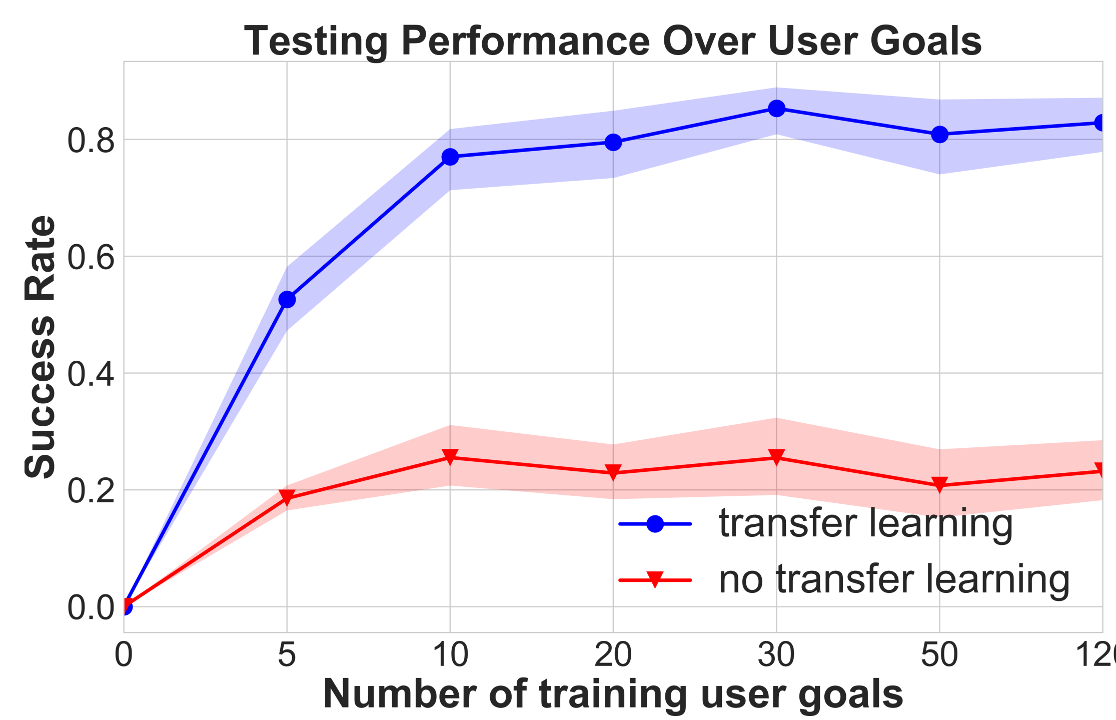




Domain Extension





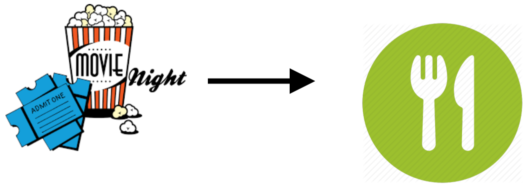


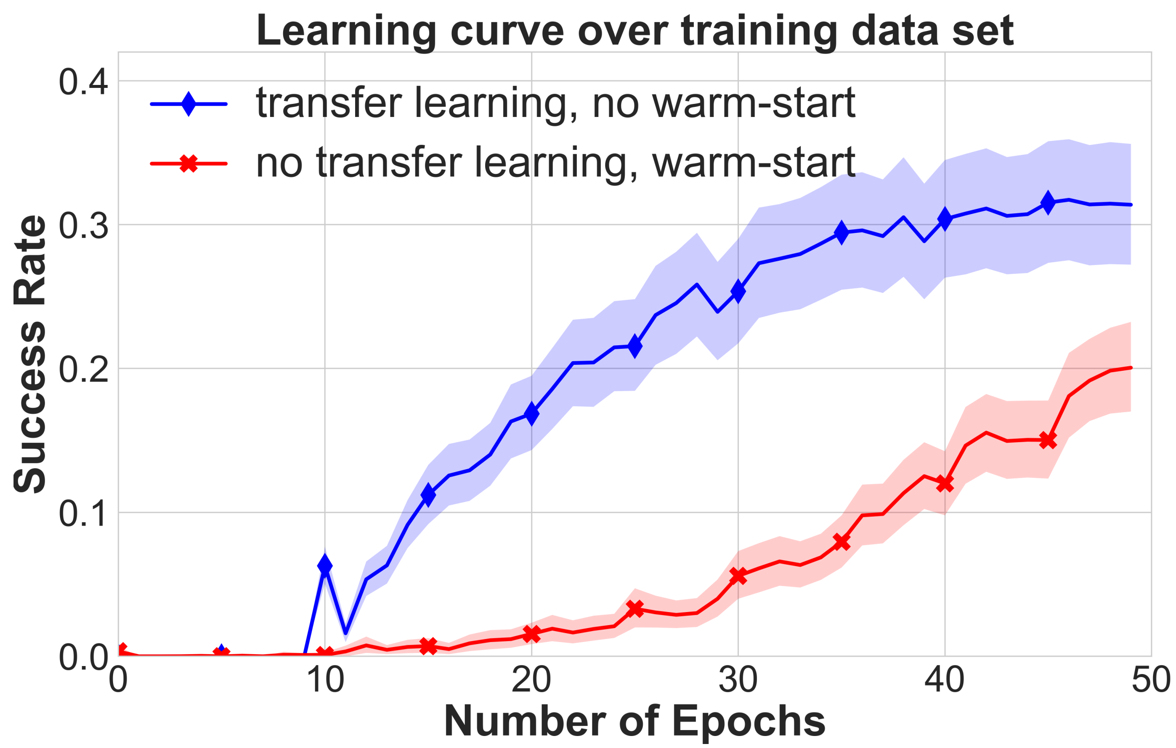
Faster Learning

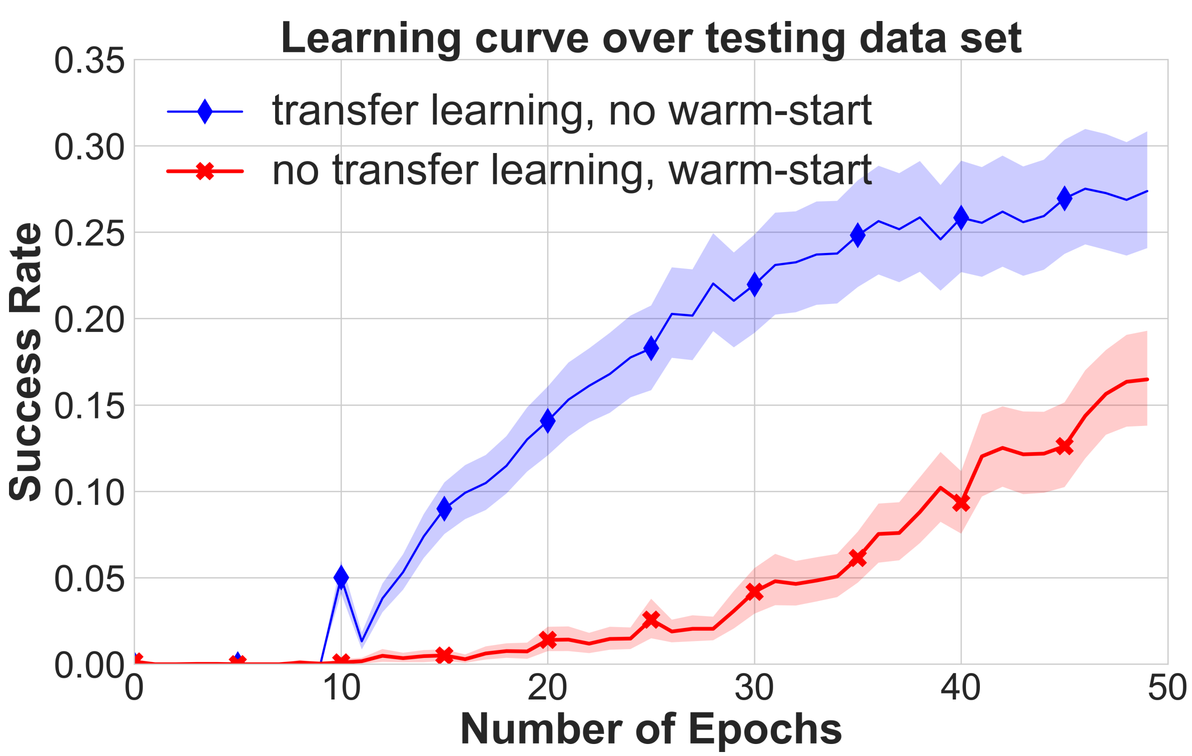
- For both models we do 100 iterations of:
  - Train using the full set of 120 user goals
  - Test on the set of 32 testing user goals
  - Transfer learning model: does not take warm-starting
  - Model from scratch: takes warm-starting
  - Report learning curve

Faster Learning - Results

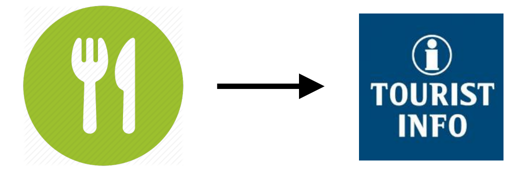
Domain Overlapping

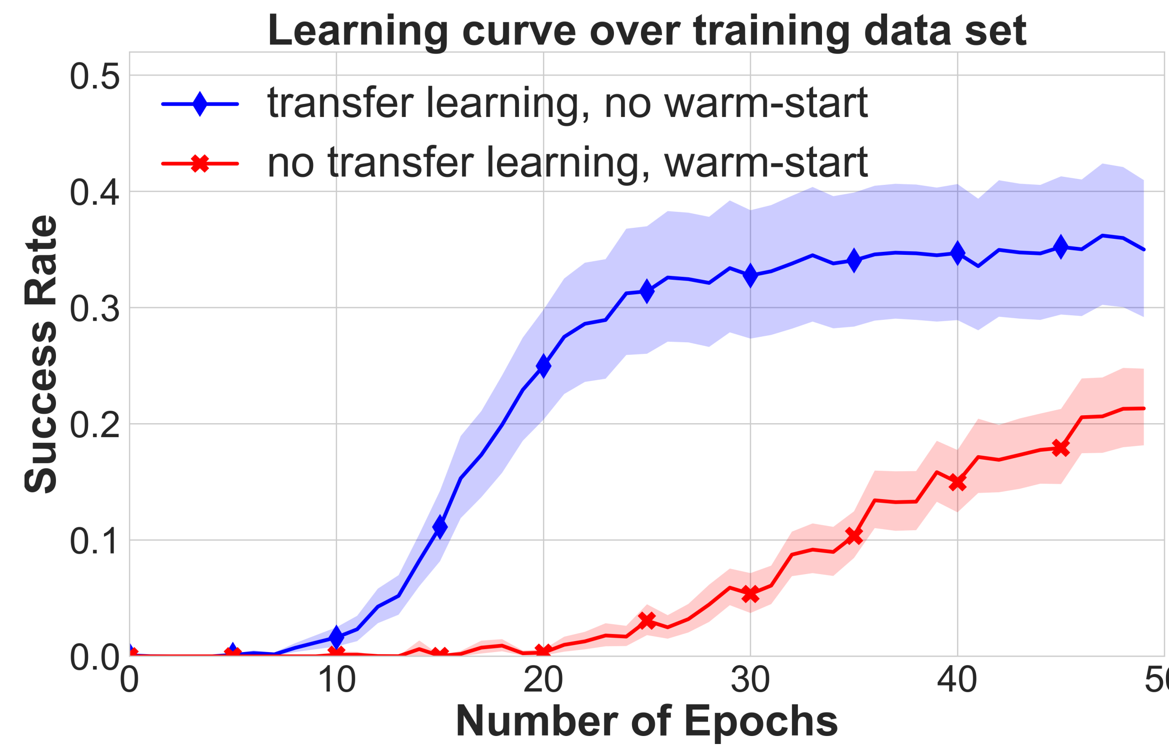


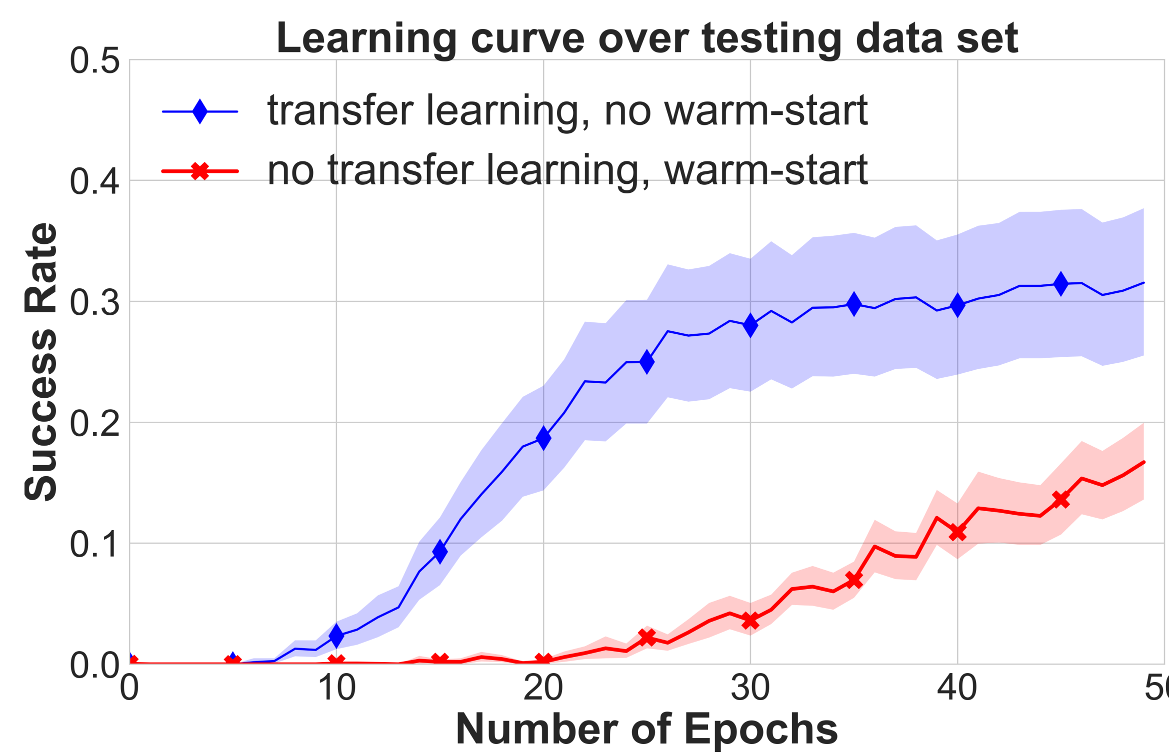




Domain Extension







Conclusion

- Training GO Chatbots with less data
- Better performances
- Faster learning

Thanks for your attention

Questions?