Domain-independent User Simulation with Transformers for Task-oriented Dialogue Systems

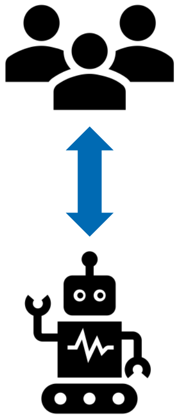
Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk,
Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gašić
Dialog Systems and Machine Learning
Heinrich Heine University Düsseldorf

A proper environment to train a dialogue system

Heinrich Heine Universität Düsseldorf

The problems of training with ...

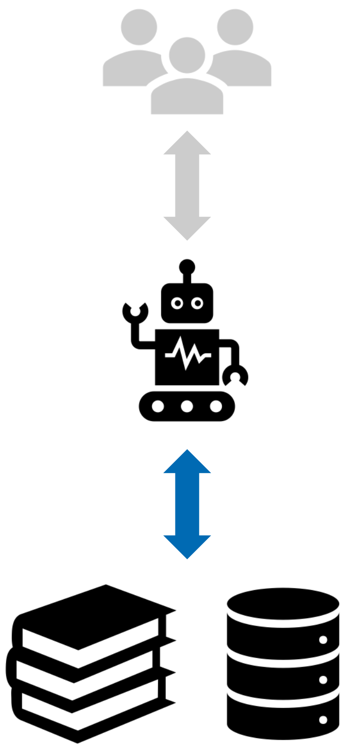
- Real users
  - Time consuming
  - Noisy feedback



A proper environment to train a dialogue system

The problems of training with ...

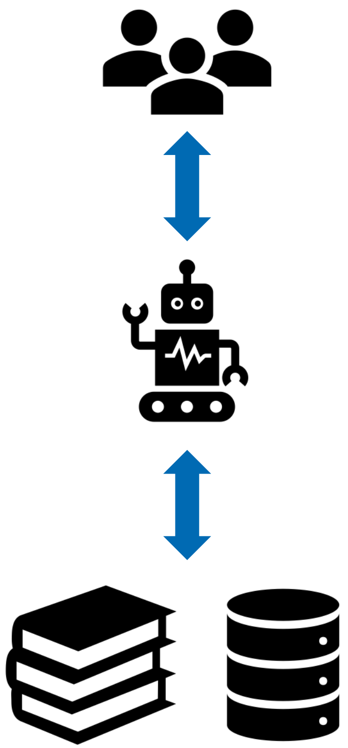
- **Real users**
  - Time consuming
  - Noisy feedback
- **Corpus**
  - Not interactive
  - Only includes a limited amount of trajectories



A proper environment to train a dialogue system

The problems of training with ...

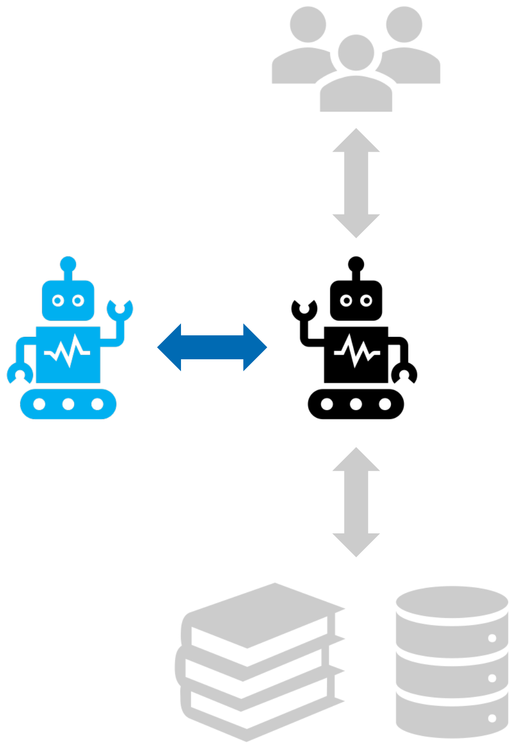
- Real users
  - Time consuming
  - Noisy feedback
- Corpus
  - Not interactive
  - Only includes a limited amount of trajectories
- Limited coverage
  - Not feasible to explore all possible paths



A proper environment to train a dialogue system

Training with ...

- User simulator
  - Efficient
  - Controllable
  - Interactive



Training with user simulators

Heinrich Heine
Universität Düsseldorf

The problems of domain-dependent models

- Rule-based user simulator (Schatzmann et al., 2007)
  - Behaviour is different to real users
  - Re-write rules when adapting to new domains
  - Design rules for complex scenarios is difficult

Training with user simulators

The problems of domain-dependent models

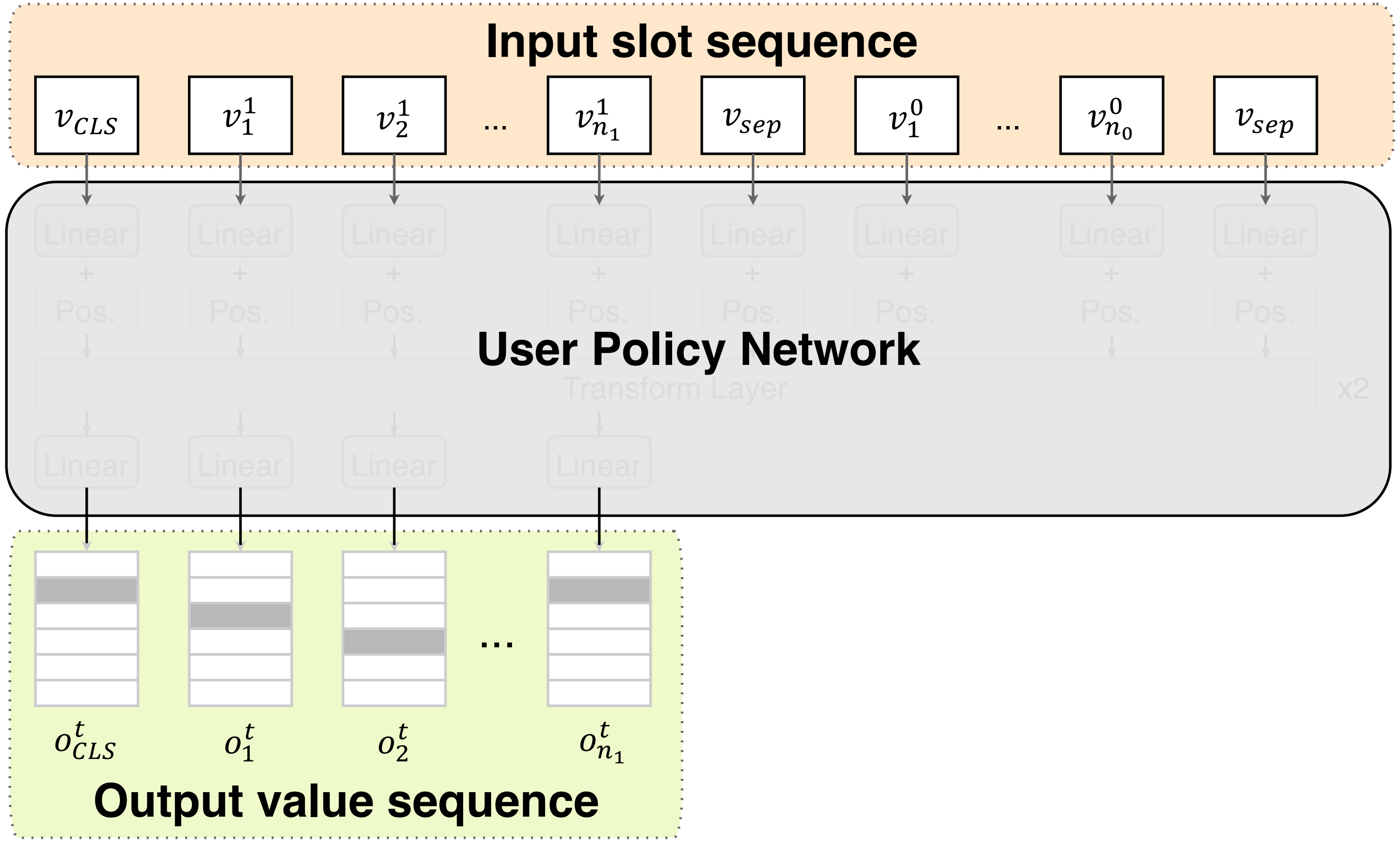
- Rule-based user simulator (Schatzmann et al., 2007)
  - Behaviour is different to real users
  - Re-write rules when adapting to new domains
  - Design rules for complex scenarios is difficult
- Statistical user simulator (Kreyssig et al., 2018, Gür et al., 2018)
  - Still domain-dependent (feature representation or output target)
  - Need new labels
  - Feature representation modification
  - Retrain the whole model

Model structure

hhu
Heinrich Heine
Universität Düsseldorf

Transformer-based domain-independent **U**ser **S**imulator (TUS)

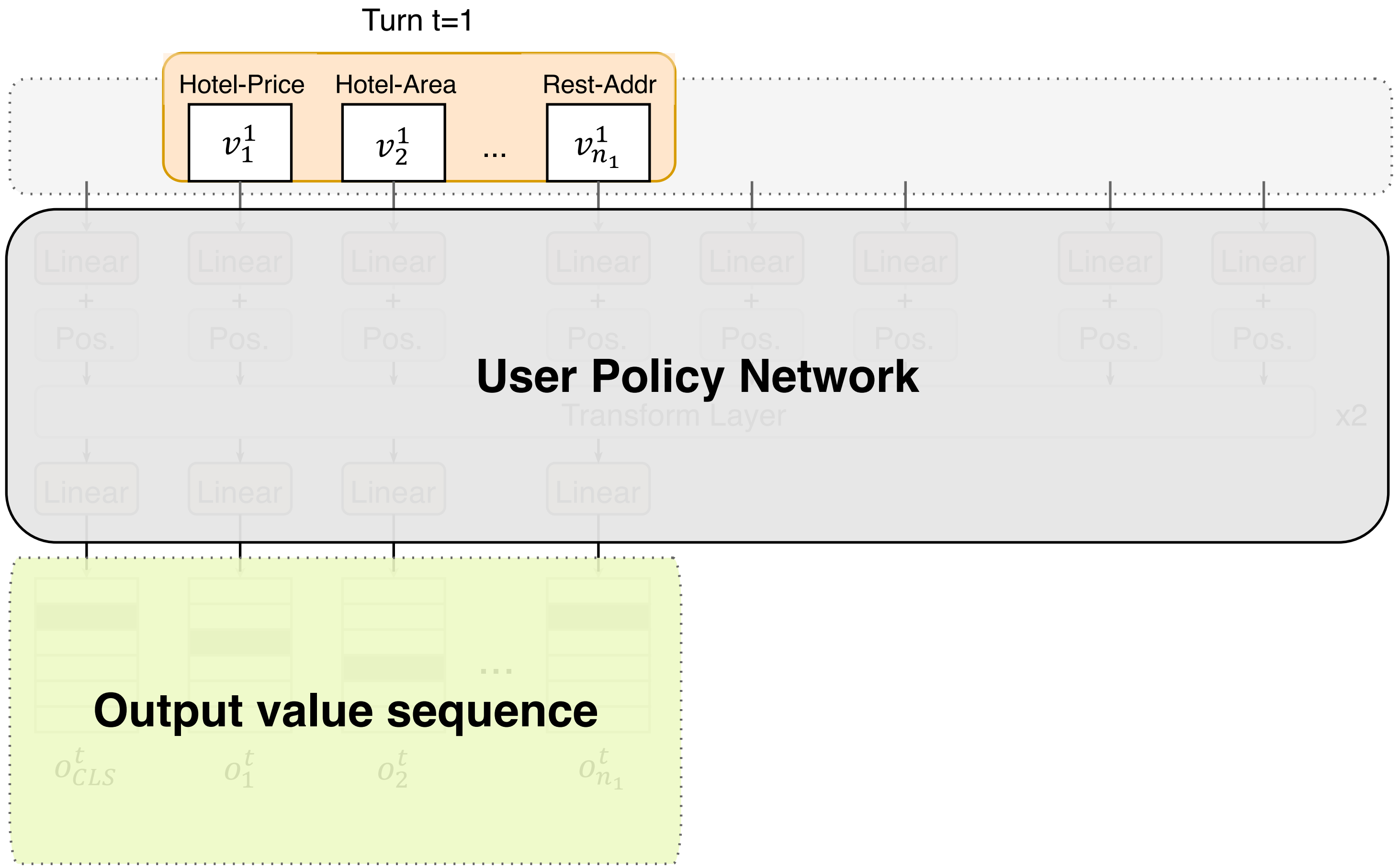
- Input: slot sequence
- Output: value sequence
- Domain-independent feature representation



Domain-independent input

Input sequence

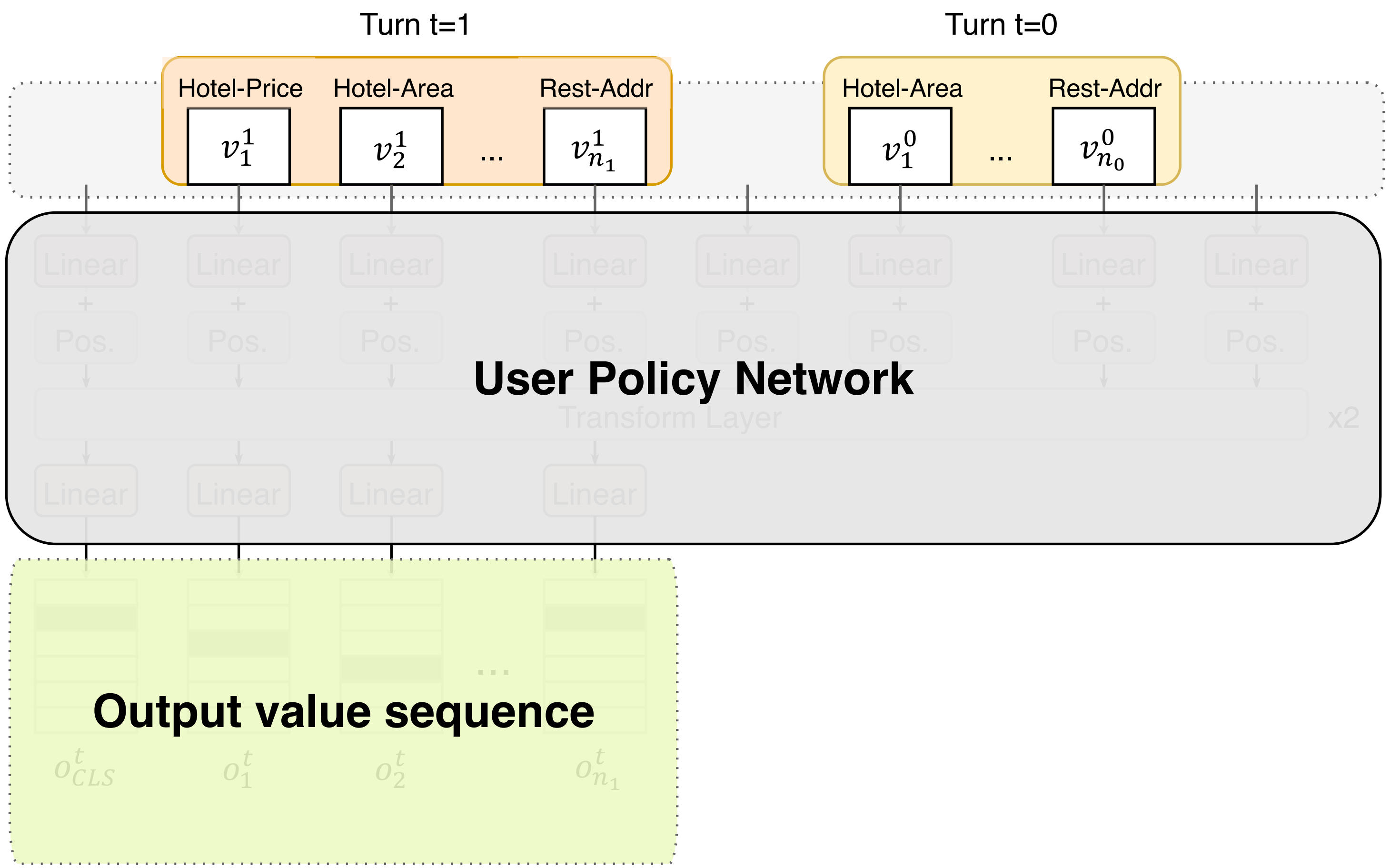
- Slots from user goal
- Slots mentioned by system
- Slot order: user’s priorities



Domain-independent input

Input sequence

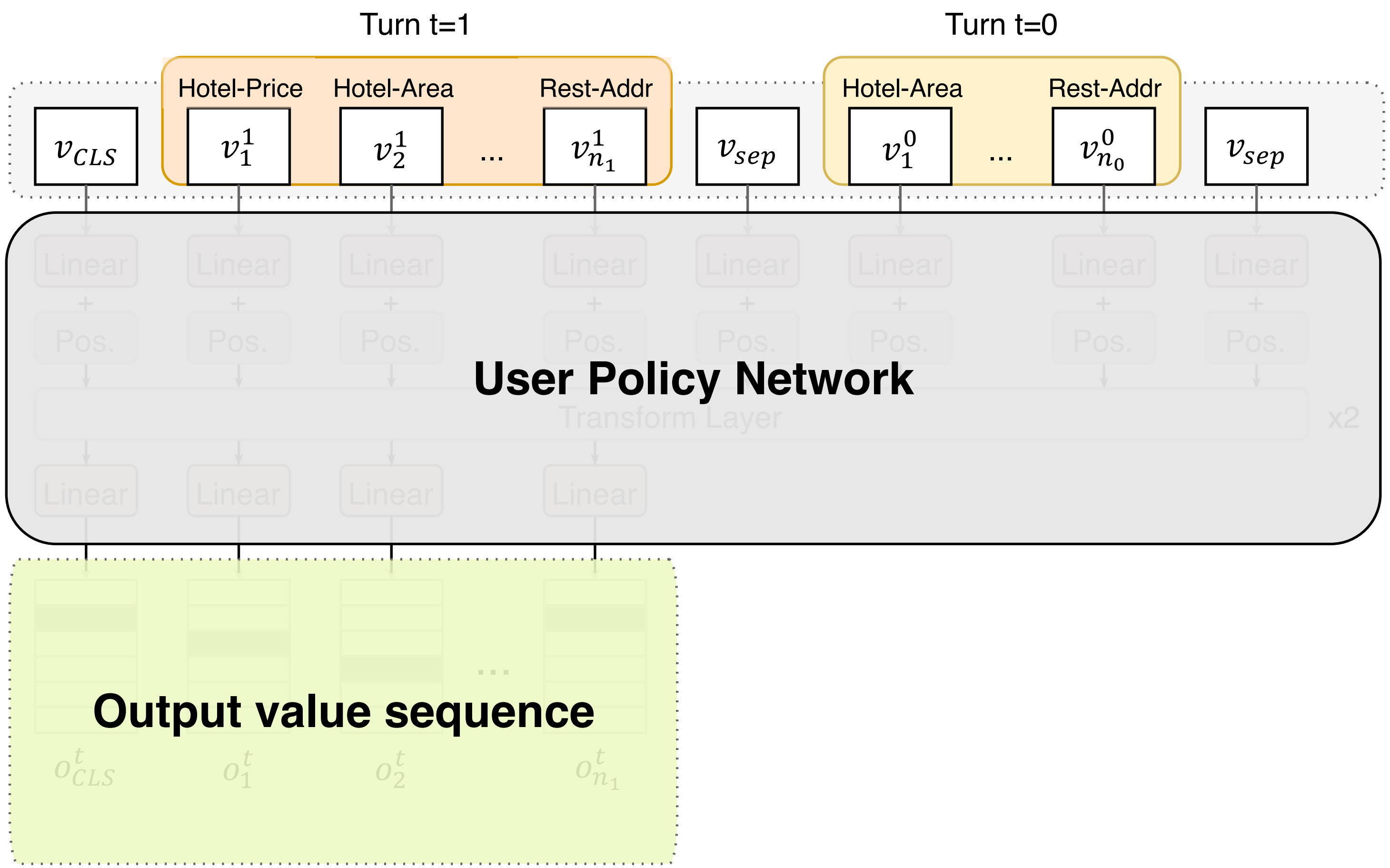
- Slots from user goal
- Slots mentioned by system
- Slot order: user's priorities
- History information



Domain-independent input

Input sequence

- Slots from user goal
- Slots mentioned by system
- Slot order: user's priorities
- History information
- Special tokens



Domain-independent input

Feature representation for each slot

- Statistical features
  - which type the slot is (inform, request...)
  - whether the slot is fulfilled
  - ...

	v_{type}	v_{ful}	
Hotel-Area	1 0	0	...

Domain-independent input

Heinrich Heine
Universität Düsseldorf

Feature representation for each slot

- Statistical features
  - which type the slot is (inform, request...)
  - whether the slot is fulfilled
  - ...
- Different slots may have the same statistical features

	v_{type}	v_{ful}	
Hotel-Area	1 0	0	...
Hotel-Price	1 0	0	...

Domain-independent input

Feature representation for each slot

- Statistical features
  - which type the slot is (inform, request...)
  - whether the slot is fulfilled
  - ...
- Different slots may have the same statistical features
- Dialogue-scope identity
  - Just for the duration of one dialogue

	v_{type}	v_{ful}		v_{index}^{domain}	v_{index}^{slot}
Hotel-Area	1 0	0	...	1 0 0 0 0	1 0 0 0 ... 0
Hotel-Price	1 0	0	...	1 0 0 0 0	0 1 0 0 ... 0

Domain-independent input

Feature representation for each slot

- Statistical features
  - which type the slot is (inform, request...)
  - whether the slot is fulfilled
  - ...
- Different slots may have the same statistical features
- Dialogue-scope identity
  - Just for the duration of one dialogue
  - It may be different in other dialogues

Dialogue 1

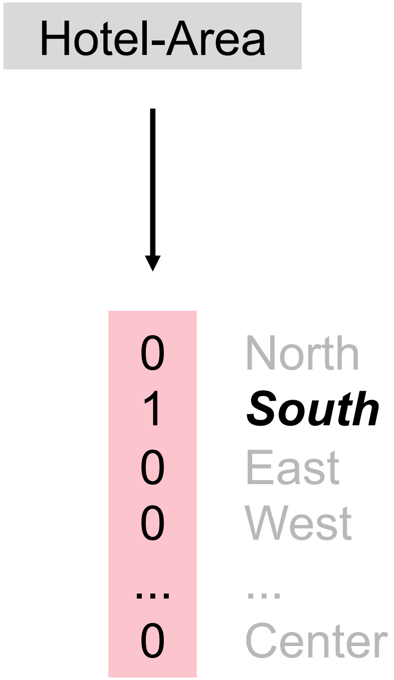
	v_{type}	v_{ful}		v_{index}^{domain}	v_{index}^{slot}
Hotel-Area	1 0	0	...	1 0 0 0 0	1 0 0 0 ... 0
Hotel-Price	1 0	0	...	1 0 0 0 0	0 1 0 0 ... 0

Dialogue 2

	v_{type}	v_{ful}		v_{index}^{domain}	v_{index}^{slot}
Hotel-Area	1 0	0	...	0 0 **1** 0 0	0 0 0 **1** ... 0
Hotel-Price	1 0	0	...	0 0 **1** 0 0	**1** 0 0 0 ... 0

Domain-independent output

- The value of each slot
  - Instead of predicting which value belongs to the slot



Domain-independent output

- The value of each slot
  - Instead of predicting which value belongs to the slot
  - TUS predicts where the value comes from

Hotel-Area

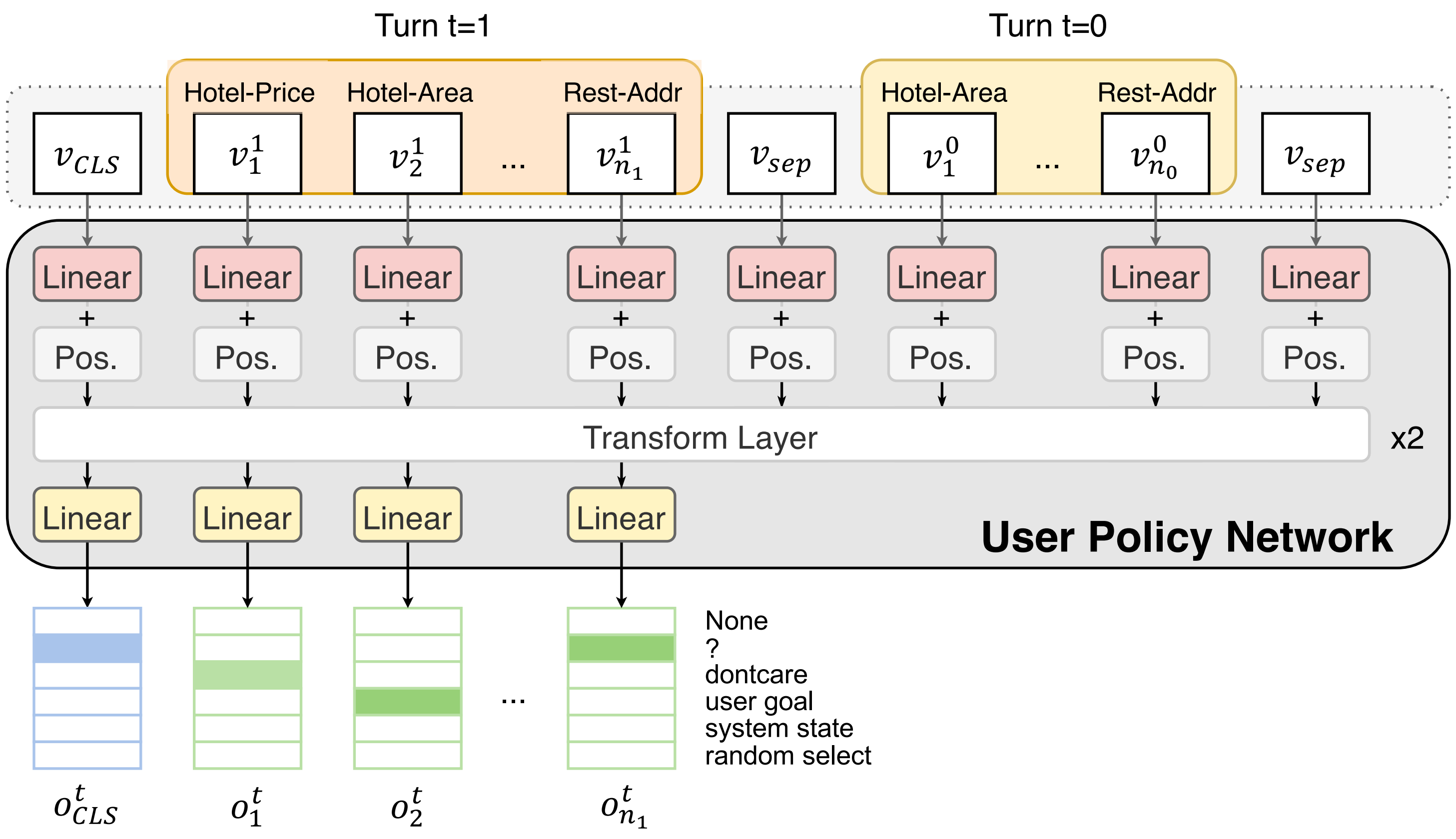
0	None
0	?
0	dontcare
1	***user goal***
0	system state
0	random select

User goal
Hotel-Area: South
Hotel-Price: Cheap
...

Domain-independent models

When adapting to new domains

- No need
  - Feature modification
  - retraining models



Experiment

Supervised training for TUS

- Dataset: MultiWOZ 2.1 (Eric et al., 2020)
- Order of the slots in the input sequence
  - Training and testing with the dataset: based on the data
  - Inference without the dataset: randomly generated

Experiment

Training Policies with TUS

- A better user simulator → a better dialogue policy
  - The performance of policies is an evaluation metric for user simulators

Experiment

Heinrich Heine
Universität Düsseldorf

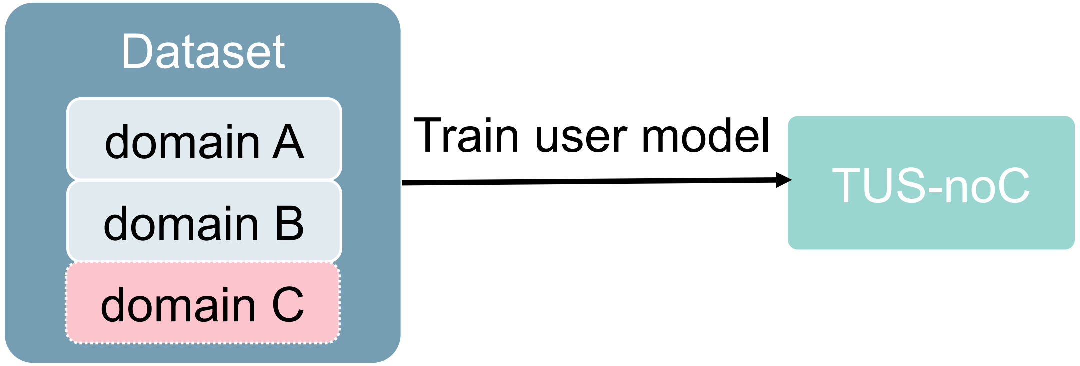
Training Policies with TUS

- A better user simulator → a better dialogue policy
  - The performance of policies is an evaluation metric for user simulators
- Policies are trained by proximal policy optimization (Schulman et al., 2017)
- Different user simulators
  - Rule-based: agenda-based user simulator (Schatzmann et al., 2007)
  - Data-driven: variational hierarchical sequence-to-sequence user simulator (Gür et al. 2018)
  - TUS

Experiment

Leave-one-domain-out Training

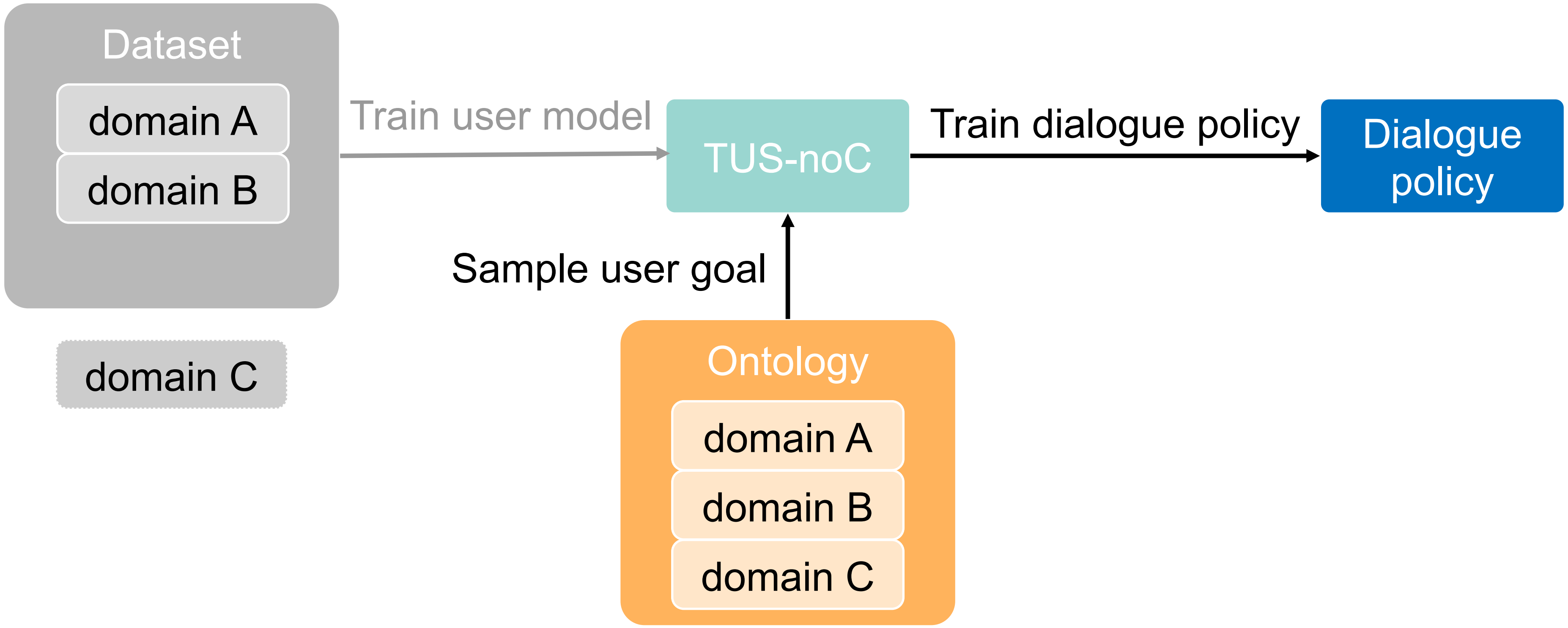
- Training TUS
  - The data related to the selected domain is removed



Experiment

Leave-one-domain-out Training

- Training dialogue policies by TUS-noX
- The user goal is sampled from all domains



Evaluation

How to evaluate a user simulator?

- Direct methods
- Indirect methods
  - Cross-model evaluation
  - Zero-shot transfer
  - Human evaluation

Cross-model evaluation

hhu
Heinrich Heine
Universität Düsseldorf

- Generalise to other user simulators
  - Trained with TUS when evaluated with ABUS: 10% absolute improvement
  - Trained with ABUS when evaluated with TUS: 35% absolute decrease

US for training	US for evaluation ABUS	VHUS	TUS	avg.
ABUS	0.93	0.09	0.58	0.53
VHUS	0.62	0.11	0.37	0.36
TUS	0.79	0.10	0.69	0.53

The success rates

Cross-model evaluation

- Generalise to multi-domain scenario
  - VHUS was designed on single-domain

US for training	US for evaluation			
	ABUS	VHUS	TUS	avg.
ABUS	0.93	0.09	0.58	0.53
VHUS	0.62	0.11	0.37	0.36
TUS	0.79	0.10	0.69	0.53

The success rates

Human evaluation

Evaluate policies trained with different user simulators in interaction with humans

- Success rate: whether the given goal is fulfilled based on the user's opinion
- Overall rate: grades the system's performance, 1 (poor) to 5 (excellent)

US for training	success			overall
	Attr.	Hotel	all	
ABUS	0.76	0.70	0.83	3.90
TUS	0.73	0.69	0.83	4.03
TUS-noAttr	0.75	0.54	0.81	4.01
TUS-noHotel	0.73	0.55	0.76	3.86

Human evaluation

Heinrich Heine Universität Düsseldorf

In comparison to ABUS, without domain-specific information ...

- Comparable success rate
- Slightly better on overall rating

US for training	success			overall
	Attr.	Hotel	all	
ABUS	0.76	0.70	0.83	3.90
TUS	0.73	0.69	0.83	4.03
TUS-noAttr	0.75	0.54	0.81	4.01
TUS-noHotel	0.73	0.55	0.76	3.86

Human evaluation

Zero-shot transfer

- The performance of TUS-noAttr is similar to the one of ABUS and TUS

US for training	success Attr.	success Hotel	success all	overall
ABUS	0.76	0.70	0.83	3.90
TUS	0.73	0.69	0.83	4.03
TUS-noAttr	0.75	0.54	0.81	4.01
TUS-noHotel	0.73	0.55	0.76	3.86

Human evaluation

Heinrich Heine Universität Düsseldorf

Zero-shot transfer

- The performance of TUS-noAttr is similar to the one of ABUS and TUS
- TUS-noHotel is worse because around 40% amount of training data is removed

US for training	success Attr.	success Hotel	success all	overall
ABUS	0.76	0.70	0.83	3.90
TUS	0.73	0.69	0.83	4.03
TUS-noAttr	0.75	0.54	0.81	4.01
TUS-noHotel	0.73	0.55	0.76	3.86

Human evaluation

Heinrich Heine
Universität Düsseldorf

Zero-shot transfer, for both TUS-noAttr and TUS-noHotel

- Comparable result on domain “attraction”, worse performance on domain “hotel”
- Domain agnostic feature, when removing one domain,
  - The success rate in the corresponding domain does not decrease
  - Domains that need plenty of data to learn are impacted

US for training	success Attr.	success Hotel	success all	overall
ABUS	0.76	0.70	0.83	3.90
TUS	0.73	0.69	0.83	4.03
TUS-noAttr	0.75	0.54	0.81	4.01
TUS-noHotel	0.73	0.55	0.76	3.86

Conclusion

Heinrich Heine Universität Düsseldorf

- TUS is domain-independent
- TUS outperforms VHUS and is comparable with ABUS
- The zero-shot transfer experiment shows that TUS can handle unseen domains without feature modification or model retraining

Conclusion

- TUS is domain-independent
- TUS outperforms VHUS and is comparable with ABUS
- The zero-shot transfer experiment shows that TUS can handle unseen domains without feature modification or model retraining
- Future work
  - Learn natural language generation
  - apply reinforcement learning to user model training

European Research Council
Established by the European Commission
Alexander von Humboldt
Stiftung / Foundation
Google Cloud

Heinrich Heine
Universität Düsseldorf

Thank you

code: https://gitlab.cs.uni-duesseldorf.de/general/dsml/tus_public

visit us at: https://www.cs.hhu.de/en/research-groups/dialog-systems-and-machine-learning.html