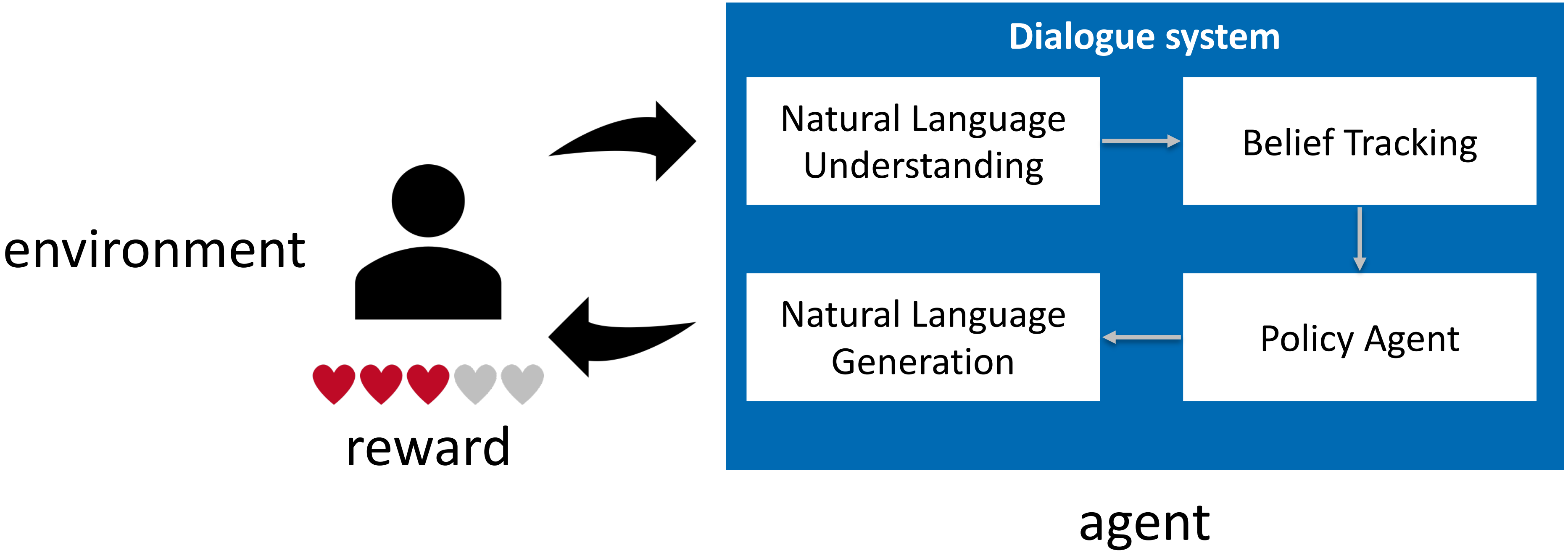
USER SIMULATION FOR DIALOGUE SYSTEMS

Hsien-Chin Lin, 22 Nov 2019

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Why do we need a simulated user (SU)?



Why do we need a simulated user (SU)?

For training

- RL need lots of interaction to learn the policy
- Learning from real user
  - costly
  - time-consuming
- Learning from data
  - collecting interactable data is not easy
- Learning from SU

Why do we need a simulated user (SU)?

For evaluation

- Human evaluation
  - costly and time-consuming
  - hard to reproduce
- Automatic evaluation
  - success rate, rewards, ...
  - NLG metrics: not consistant with human evaluation
- Evaluating by SU is easy to reproduce, cross-model comparison

Different kinds of user simulation

Summarize SU in different aspects

- Granularity
  - Semantic level
  - Natural Language level
    - template, retrieval, generation
- Methodology
  - n-gram: Bi-gram, graph model, bayesian model, HMM, ...
  - rule-based: agenda-based
  - data driven: Seq2Seq, inverse RL, adversarial model, ...

Previous studies

non-DL approaches

- N-gram
- Graph based
- Agenda based

N-grams SU (Eckert et al. 1997)

- Bi-gram model $P(a_u|a_m)$
  - only looks on the latest system action
  - cannot produce coherent user behavious
  - the SU may produce illogical behaviour if the user goal changes
- Look longer history
- incorporate user goal into user state
- HMM (Cuayáhuitl et al. 2005), Baysian model (Pietquin and Dutoit 2009)...

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Graph-based SU (Scheffler and Young, 2000)

- All possible paths in a network
- Need extensive domain knowledge
- Not practicable for complex domain

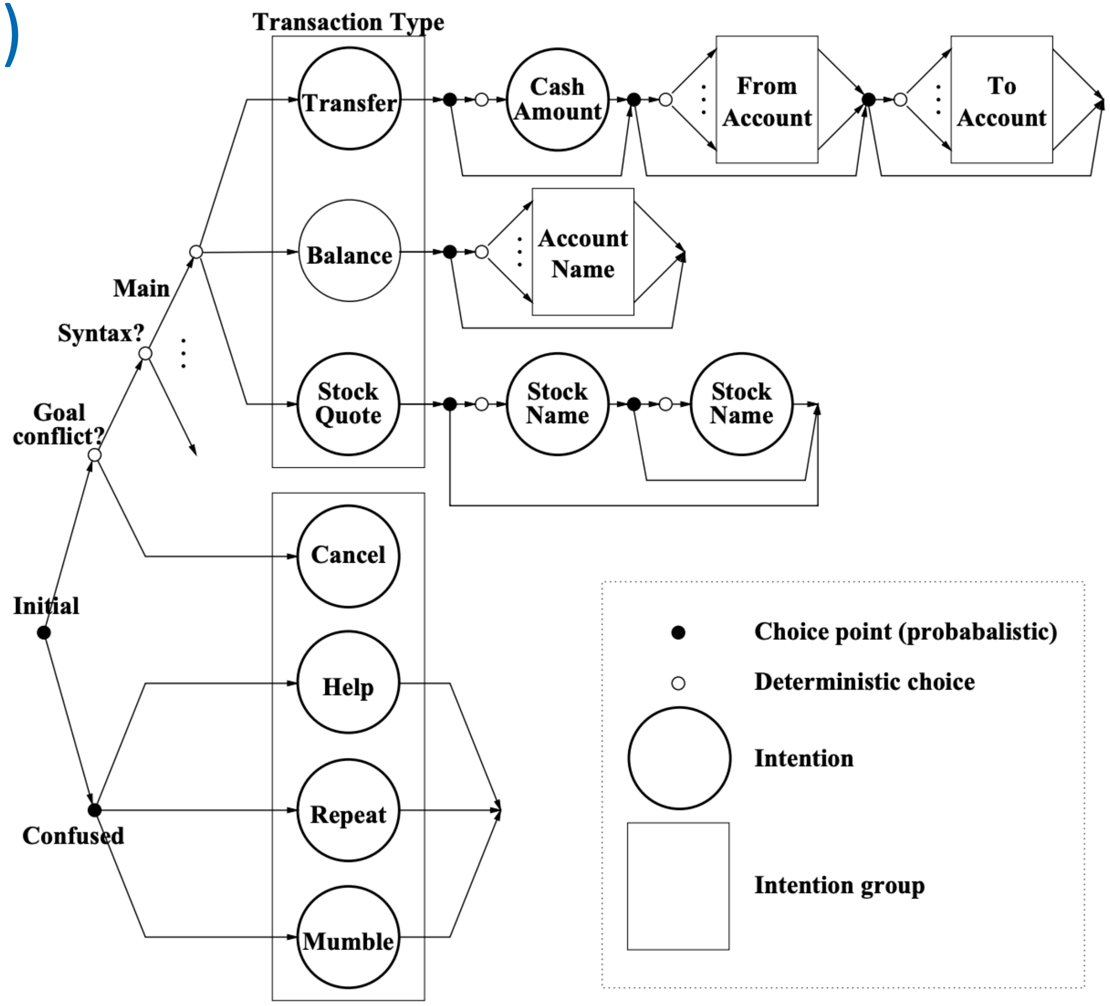


Figure 3: Partial structure for utterance construction in the banking application.

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Agenda-based approach (Schatzmann et al. 2007)

- user state S is described as an agenda A and a goal G
- Example:

$$C_0 = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \end{bmatrix}$$

$$R_0 = \begin{bmatrix} name = \\ addr = \\ phone = \end{bmatrix}$$

Sys 0	Hello, how may I help you?

$$A_1 = \begin{bmatrix} inform(type = bar) \\ inform(drinks = beer) \\ inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Usr 1	I'm looking for a nice bar serving beer.
Sys 1	Ok, a wine bar. What pricerange?

$$A_2 = \begin{bmatrix} negate(drinks = beer) \\ inform(pricerange = cheap) \\ inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Usr 2	No, beer please!

- The probabilities can be learned from corpus or set manually

Summary of these models

These models suffer from...

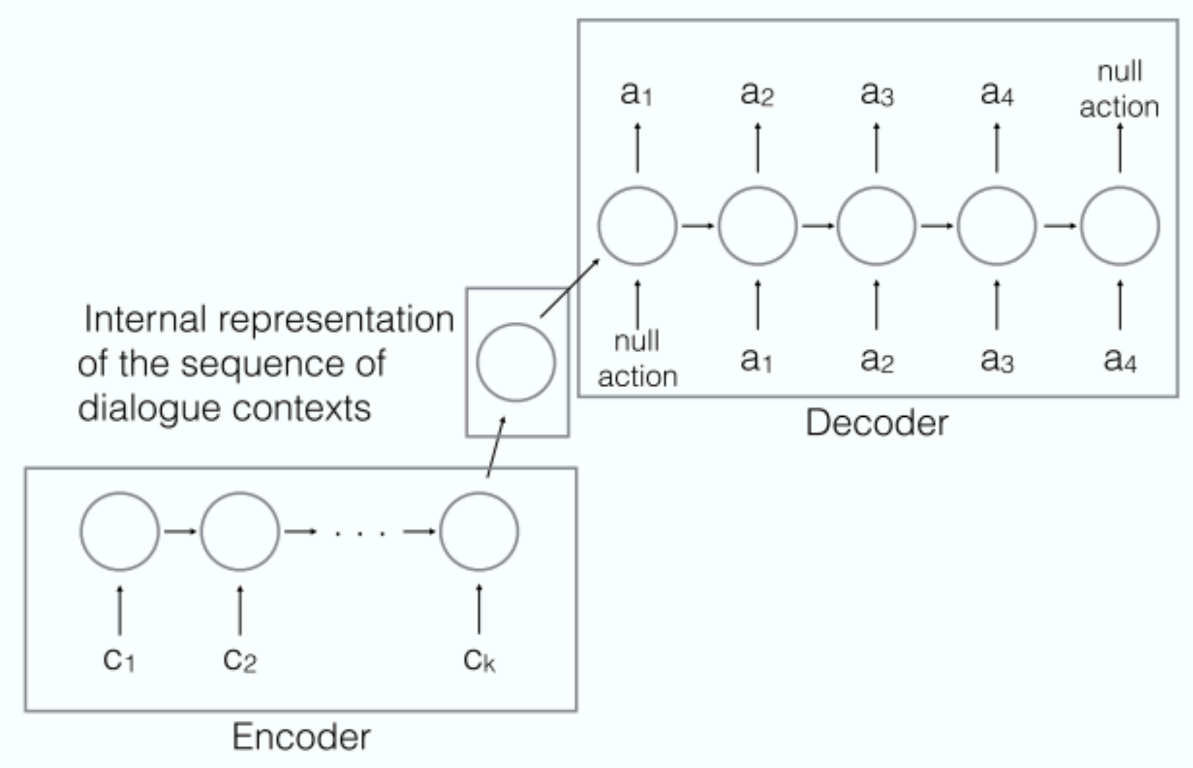
- Inability to take dialogue history
- Rigid structure to ensure coherent user behavior
- Need lots of labor effort for designing rules
- Domain dependent

Seq2Seq models

- Semantic to Semantic
- Combined agenda-base with seq2seq
- Semantic to Utterence
- Hierarchical seq2seq
- comparison of different settings

semantic level (El Asri et al., 2016)

- uniform select a goal $G = (C, R)$
  - C: *constraints*, food-type, price range, ...
  - R: *requests*, name, address, ...
- context c_t concatenated with
  - $a_{m,t}$: recent machine acts
  - $inconsist_t$: inconsistency
  - $const_t$: constraints status
  - req_t: requests status



HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Example of the context vector

Machine output / User answer	Machine acts	Inconsistency vector	Constraints status	Requests status
Welcome! How may I help you?	0000000010 greet	000000	001	10110111
Is there a cheap restaurant downtown?				
A cheese restaurant. What is your budget?	0000010001 implicit-confirm, request	000010	011	10110111
No, I said a cheap restaurant.				
Panda express is a cheap restaurant downtown.	0100000100 offer, inform	000000	111	10110111
What is the address of this place?				
Panda express is located at 108 Queen street.	0100000100 offer, inform	000000	111	10111111

Table 1: Examples of contexts in a dialogue with a restaurant-seeking system. The user goal has two constraints (cheap and downtown) and one request (address).

Experiment

- Dataset: DSTC2, DSTC3
- Baseline
  - Bi-gram, agenda-based
  - Sequence-to-one:
    outputs a probability distribution over a predefined set of compound acts (size: 54)
- Measurement
  - F-score, i.e. $precision = \frac{\#\,of\ correctly\ predicted\ dialog\ acts}{\#\,of\ predicted\ dialog\ acts}$

Result

- Average F-score on 50 runs

Dataset	Bigram	Agenda-based	Sequence-to-one	Sequence-to-sequence
DSTC2 Validation	0.20	0.24	0.37	0.34
DSTC2 Test	0.09	0.18	0.29	0.27
DSTC3 Test	—	0.13	0.19	0.18

- The Seq2One is slightly better than Seq2Seq because it‘s an easier task
- The Seq2Seq has better scalability (the number of possible acts might grow)
- The recall is relatively low on larger actions space (54 in DSTC2, 94 in DSTC3)

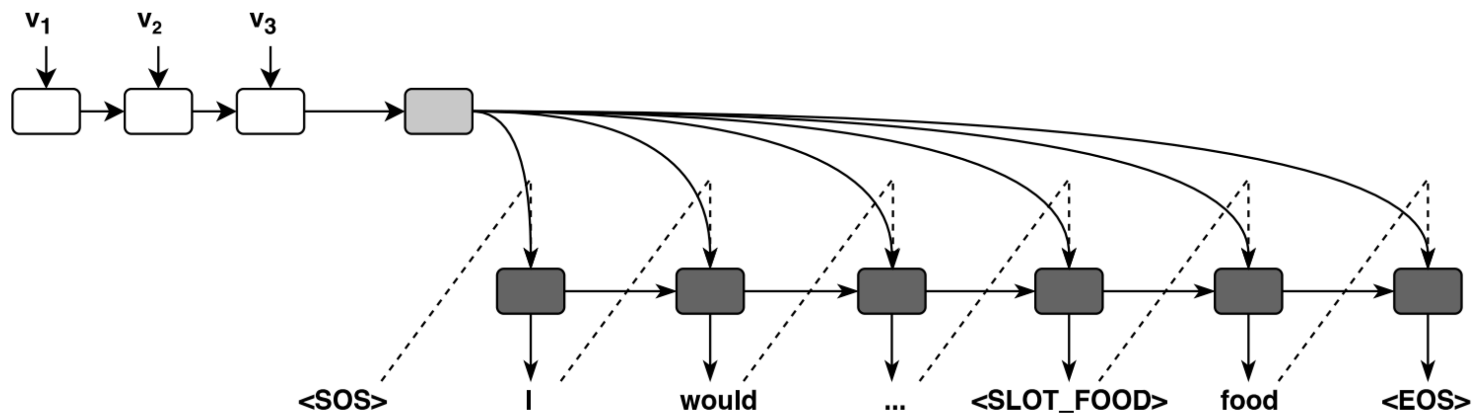
Combined agenda-based model with Seq2Seq model (Xiujun Li et al. 2017)

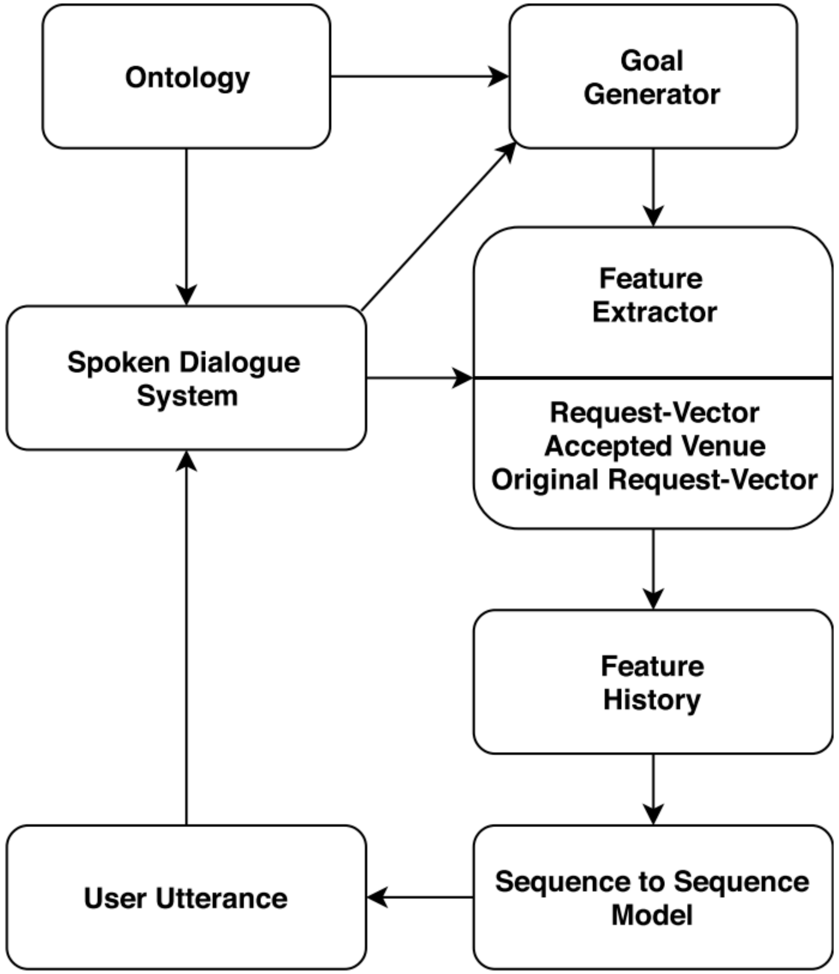
- Use the agenda-based model for planning
- If the dialog act can be found in templates then use templates
- Else use Seq2Seq model for NLG

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Semantic to Utterance (Kreyssig et al. 2018)

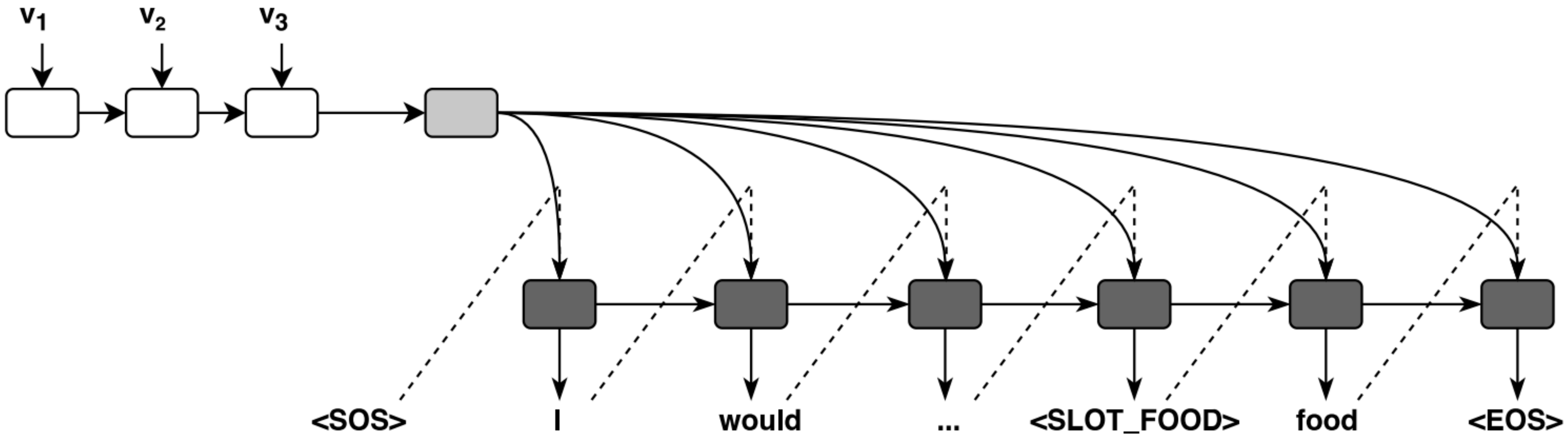
- System structure
  - The setting of Goal Generator and Feature Extractor is like (El Asri et al., 2016)
  - The input sequence is Feature History
  - The output seqence is User Utterance





Generate non-deterministic result

- Beam-search is often used to generate a sequence by RNNs
- Taking n beams with the highest probability $P(w_t w_{t-1} \dots w_0 | \boldsymbol{p})$



- Sample n words per beam from the probability distribution

Experiments – Cross-Model Evaluation

- The policy trained with NUS can perform well on both SUs
- Overfitting: the policy performing best on the NUS was not the one on the ABUS

Train. Sim.	Eval. Sim.			
	NUS		ABUS	
	Rew.	Suc.	Rew.	Suc.
NUS-best	13.0	$98.0^{\mathcal{N}_1}$	13.3	99.8
ABUS-best	1.53	$71.5^{\mathcal{A}_1}$	13.8	$99.9^{\mathcal{A}_2}$
NUS-avg	12.4	96.6	11.2	94.0
ABUS-avg	-7.6	45.5	13.5	99.5

Table 2: Results for policies trained for 4000 dialogues on NUS and ABUS when tested on both USs for 1000 dialogues. Five policies with different initialisations were trained for each US. Both average and best results are shown.

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Experiments – Cross-Model Evaluation

- In five seeds for NUS, the performance is all better with less data
- This behavior was not observed for the policies trained with the ABUS

Train. Sim.	Eval. Sim. NUS Rew.	NUS Suc.	ABUS Rew.	ABUS Suc.
NUS-best	13.0	$98.0^{\mathcal{N}_1}$	13.3	99.8
ABUS-best	1.53	$71.5^{\mathcal{A}_1}$	13.8	$99.9^{\mathcal{A}_2}$
NUS-avg	12.4	96.6	11.2	94.0
ABUS-avg	-7.6	45.5	13.5	99.5

Table 2: Results for policies trained for 4000 di-

Train. Sim.	Eval. Sim. NUS Rew.	NUS Suc.	ABUS Rew.	ABUS Suc.
NUS-best	12.2	95.9	13.9	$99.9^{\mathcal{N}_2}$
ABUS-best	-4.0	54.8	13.2	99.0
NUS-avg	12.0	95.4	12.2	97.3
ABUS-avg	-9.48	42.3	12.8	98.4

Table 3: As Table 2 but trained for 1000 dialogues.

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Experiments – human Evaluation

- The NUS performs better
- The overfitting is also observed, the best performing policy was the policy that performed best on the other US

Training Simulator	Human Evaluation	
	Rew.	Suc.
NUS - $\mathcal{N}_1$	13.4	91.8
NUS - $\mathcal{N}_2$	13.8	93.4
ABUS - $\mathcal{A}_1$	13.3	90.0
ABUS - $\mathcal{A}_2$	13.1	88.5

Table 4: Real User Evaluation. Results over 250 dialogues with human users. $\mathcal{N}_1$ and $\mathcal{A}_1$ performed best on the NUS. $\mathcal{N}_2$ and $\mathcal{A}_2$ performed best on the ABUS. Rewards are not comparable to Table 2 and 3 since all user goals were achievable.

Discussion

- Less labelling for generate natural language compared with semantic response
- NUS excelled on both evaluation tasks

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Hierarchical User Simulator (HUS) (Gür et al. 2018)

- An end-to-end hierarichical seq2seq approach
- Without any feature extraction and external state tracking annotations
- Encode user goal: $h^C = Enc(e^C; \theta_C)$
- Encode system turn: $h_i^S = Enc(e^{S_i}; \theta_S)$
- Encode dialogue history
  $h_0^D = h^C$
  $h_i^D = Enc(\{h_i^S\}_{i=1}; \theta_D)$
- $L_{crossent}$: cross-entropy error between candidate and correct user sequence

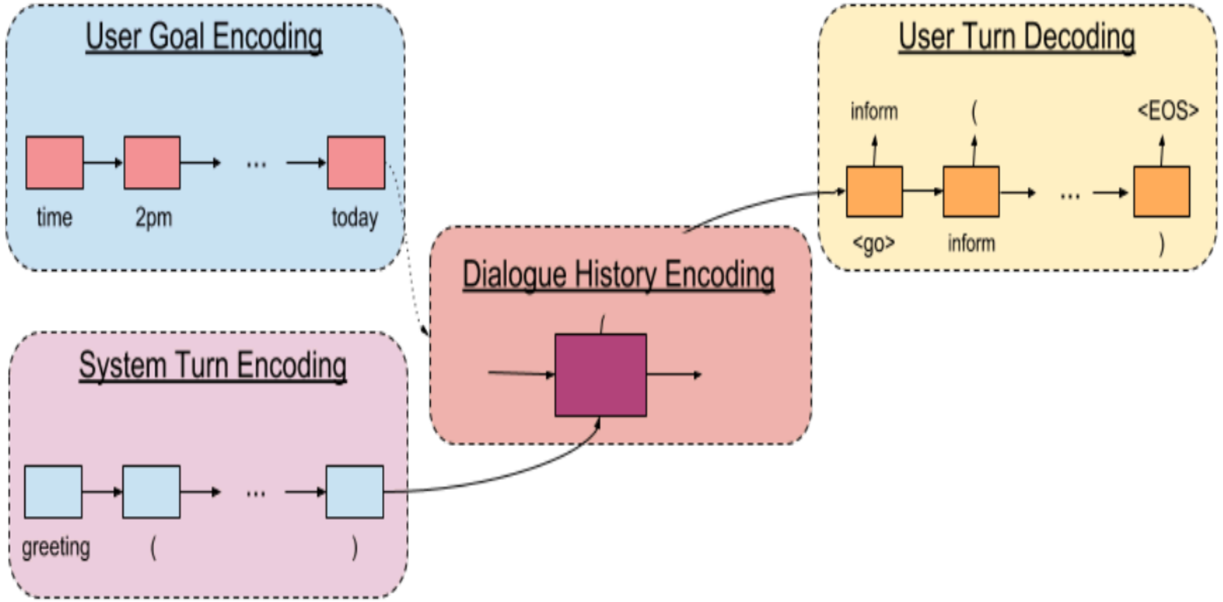
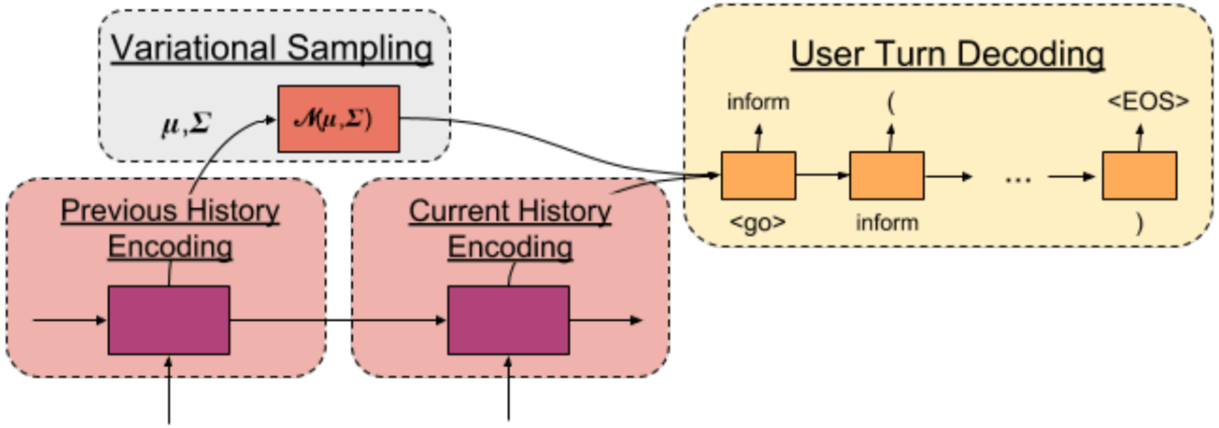


Fig. 2: HUS model: Boxes are RNN cells, colors indicate parameter sharing.

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Variational HUS (VHUS)

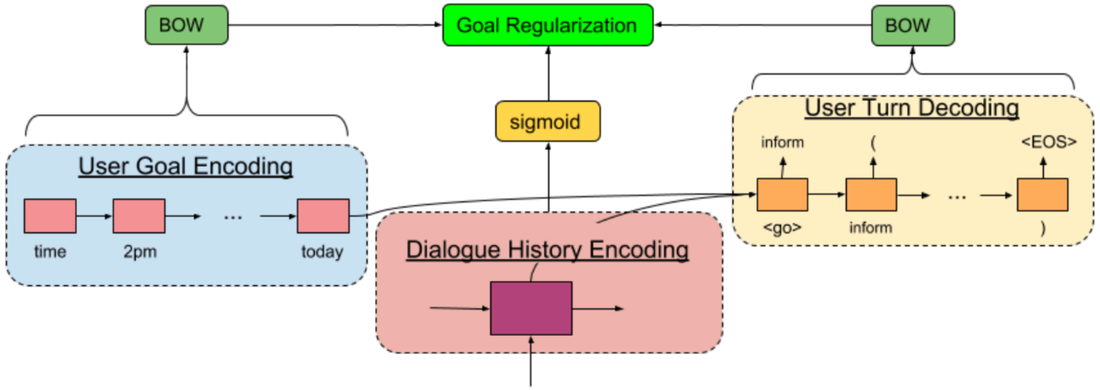


- The output of HUS is deteministic
- Add a Gaussian distribution generator
- Sample $z_x \sim N(z|\mu_x, \Sigma_x)$
  $\mu_x = W_\mu h_{t-1}^D + b_\mu$
  $\Sigma_x = W_\Sigma h_{t-1}^D + b_\Sigma$
- The decoder will be initialized with $\hat{h}_t^D = FC([h_t^D; z_x])$
- KL divergence between prior and posterior distribution
  $L_{var} = \alpha KL\big(N(z|\mu_x, \Sigma_x)|N(z|\mu_y, \Sigma_y)\big)$
  in order to make sure the behavior will be consistent

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Goal Reqularization (VHUSReg)

- Generating long dialogues when user turns diverge from the initial user goal
- Initialize the history encoder with zero, then $\hat{h}_t^D = FC([h_t^D; h^c])$
- Minimize the divergence between user goal and user turn token



$$L_{reg} = ||b_t^u - BOW(C)|| + ||b_t^D - BOW(U_t)|| + ||b_t^S - BOW(S_t)||$$

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Experiment results

- SL
  - Supervised end-to-end policy
  - Map user utterence to system actions
- RL policy outperformed SL
  - Especially on EM, the SL may stuck in local minima and cannot recover some of the slot-value pairs
- RL is more robust, even with weaker SU

	Exact Match (%)	Partial Match(%)	Dialogue Length	
HUS	75.67	94.3	12.03	**SL**
	94.69	98.27	7.45	**RL**
+ dialogue length	86.1	96.51	9.615	**SL**
	94.33	98.2	7.076	**RL**
VHUS	82.52	95.69	11.8005	**SL**
	95.53	98.43	7.803	**RL**
HUSReg	88.8	97.08	7.92	**SL**
	96.19	**98.56**	**6.878**	**RL**
VHUSReg	91.90	97.67	8.0555	**SL**
	95.98	98.52	6.905	**RL**

Human evaluation

- The dialogue is tranfered to natural language by template
- All SUs get better score and less standard deviation

Model	Average Score (Standard Deviation)
Agenda-based	4.56 (0.859)
HUS+dialogue length	4.86 (0.545)
VHUS	4.88 (0.472)
HUSReg	4.88 (0.452)
VHUSReg	4.83 (0.594)

Comparison between different settings (Shi et al. 2019)

- Compare different settings
  - Policy: agenda-based and model-based
  - NLG: template, retrieval, and generation
  - Evaluation: direct and indirect

Automatic direct evaluation

- Use perplexity, vocabulary size and utterence length to measure NLG quality
- Retrieval-based models have the largest Vocab
- Retrieval-based model can generate the longest sentences, but End-to-End model is also doing good
- Although the PPL is the largest for retrieval-based models, it also has the biggest Vocab and longest utterence length

Simulators	NLU	DM	NLG	PPL	Vocab	Utt	Hu.Fl	Hu.Co	Hu.Go	Hu.Div	Hu.All
Agenda-Template (AgenT)	SL	Agenda	Template	10.32	180	9.65	4.07	4.56	4.88	2.4	4.50
Agenda-Retrieval (AgenR)	SL	Agenda	Retrieval	33.90	**383**	**11.61**	3.50	4.22	4.58	3.9	3.74
Agenda-Generation (AgenG)	SL	Agenda	Generation	**7.49**	159	8.07	3.32	3.92	4.64	2.5	3.36
SL-Template (SLT)	SL		Template	9.32	192	9.83	**4.80**	**4.80**	**4.98**	2.6	**4.74**
SL-Retrieval (SLR)	SL		Retrieval	29.36	346	11.06	4.40	3.99	4.88	**4.3**	4.01
SL-End2End (SLE)	End-to-End			13.47	205	10.95	3.32	2.62	3.18	2.7	2.64

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Human direct evaluation

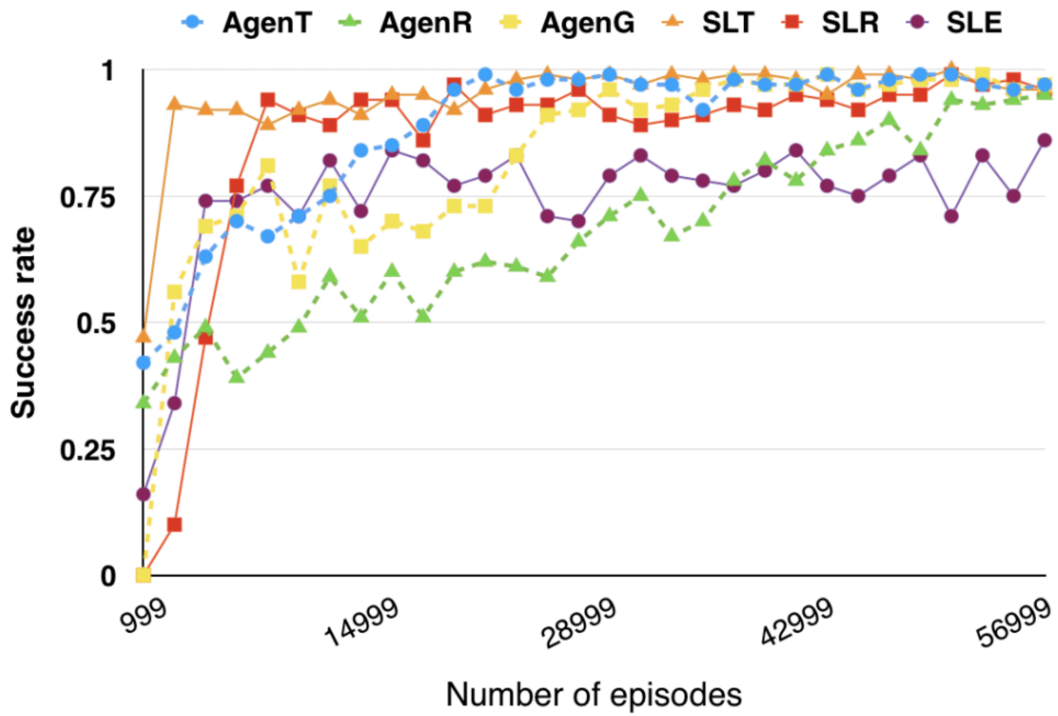
- Fluency: Templates. They are written by human
- Coherence: Agenda-based in general better than model-based
- Goal adherence: Infusing the goal is more difficult for End2End.
- Diversity: Retrieval-based is good at diversity but is not as good in fluency
  Template-based outperformed on fluency but suffer from diversity
  Generation-based suffer from generic responses

Simulators	NLU	DM	NLG	PPL	Vocab	Utt	Hu.Fl	Hu.Co	Hu.Go	Hu.Div	Hu.All
Agenda-Template (AgenT)	SL	Agenda	Template	10.32	180	9.65	4.07	4.56	4.88	2.4	4.50
Agenda-Retrieval (AgenR)	SL	Agenda	Retrieval	33.90	**383**	**11.61**	3.50	4.22	4.58	3.9	3.74
Agenda-Generation (AgenG)	SL	Agenda	Generation	**7.49**	159	8.07	3.32	3.92	4.64	2.5	3.36
SL-Template (SLT)	SL		Template	9.32	192	9.83	**4.80**	**4.80**	**4.98**	2.6	**4.74**
SL-Retrieval (SLR)	SL		Retrieval	29.36	346	11.06	4.40	3.99	4.88	**4.3**	4.01
SL-End2End (SLE)	End-to-End			13.47	205	10.95	3.32	2.62	3.18	2.7	2.64

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Automatic indirect evaluation

- Model-based converge faster. Capture the major path instead of exploring all the possible paths
- Retrieval-based converged slower because of larger vocabulary size



HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Human indirect evaluations

- The system can handle more language variations will do better on Solved ratio
- The efficiency doesn’t always correlated to the dialog length (AgenG and SLE)
- The satisfaction is not only related to solved ration but also efficiency and latency
- Naturalness is related to solved ratio (overall performance)

RL System	Solved Ratio	Satisfaction	Efficiency	Naturalness	Rule-likeness	Dialog Length	Auto Success
Sys-AgenT	0.814 ±0.06	4.29 ±0.20	4.35 ±0.21	3.96 ±0.23	4.49 ±0.15	8.95 ±0.38	**0.983** ±0.01
Sys-AgenR	**0.906** ±0.05	**4.52** ±0.15	4.45 ±0.16	4.23 ±0.19	4.59 ±0.14	**8.73** ±0.31	0.925 ±0.02
Sys-AgenG	0.904 ±0.05	4.38 ±0.18	**4.46** ±0.19	**4.33** ±0.17	4.51 ±0.16	9.48 ±0.45	0.980 ±0.01
Sys-SLT	0.781 ±0.07	3.87 ±0.22	3.81 ±0.22	3.63 ±0.22	**4.08** ±0.21	9.61 ±0.76	0.978 ±0.01
Sys-SLR	0.823 ±0.05	4.23 ±0.20	4.20 ±0.10	3.99 ±0.20	4.42 ±0.17	8.92 ±0.70	0.965 ±0.01
Sys-SLE	0.607 ±0.06	3.42 ±0.22	3.41 ±0.23	3.59 ±0.20	4.22 ±0.20	9.44 ±0.69	0.798 ±0.03

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Cross model evaluation

- Agenda-based with retrieval-based NLG has the best performance
  This result agrees with the human evaluation
- More type of SU will give better quality of evaluation
  User SLT prefers SLT (0.975) than AgenG (0.965), but in overall AgenG is better
- The diagnal is usuall the highest. RL policy is not general over all kind of users

Usr\Sys	**Sys-AgenT**	**Sys-AgenR**	**Sys-AgenG**	**Sys-SLT**	**Sys-SLR**	**Sys-SLE**
AgenT	0.975	0.960	0.790	0.305	0.300	0.200
AgenR	0.540	0.900	0.785	0.230	0.230	0.235
AgenG	0.725	0.975	0.950	0.355	0.300	0.20
SLT	0.985	0.985	0.985	0.990	0.965	0.730
SLR	0.925	0.975	0.965	0.975	0.935	0.630
SLE	0.770	0.820	0.815	0.840	0.705	0.770
Average	0.820	**0.935**	0.882	0.616	0.573	0.461

Discussion

- Model-based perform relatively worse
- Model-based doesn’t explor all possible paths (Act6)

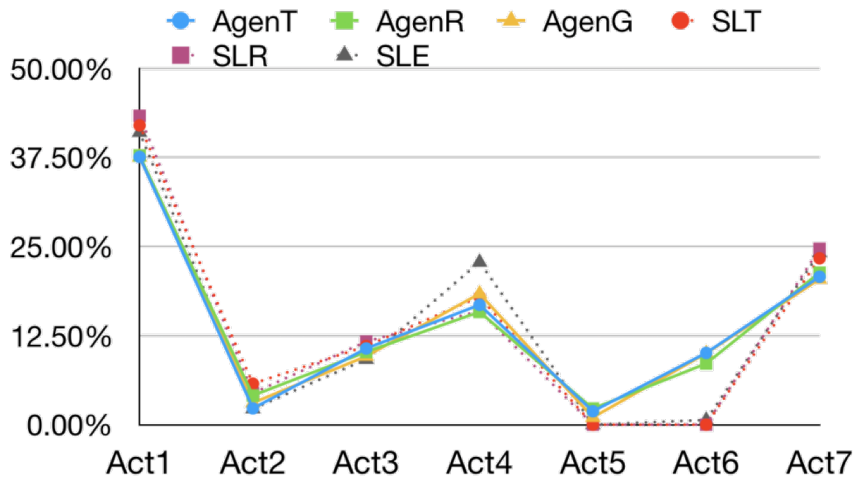


Figure 5: Dialog act distribution comparison. Act1 to Act7 corresponds to the seven user dialog acts, “*inform restaurant type*”, “*inform restaurant type change*”, “*ask info*”, “*make reservation*”, “*make reservation change time*”, “*anything else*”, and “*goodbye*”

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Summary

- The generating model may suffer from generating generatic results
- We can get better policy with more diverse output SU
- The policy of SU need to explore all possiblities

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Inverse RL (Chardramohan et al., 2011)

- The SU can be view as an MDP $\{S, A, P, \gamma\}/R$
- Reward function $R_\theta(s,a) = \theta^T \phi(s,a) = \sum_{i=1}^{k} \theta_i \phi_i(s,a)$
- Q-function $Q^\pi(s,a) = E\left[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a\right]$
- $Q^\pi(s,a) = E\left[\sum_{i=0}^{\infty} \gamma^i \theta^T \phi(s,a) | s_0 = s, a_0 = a\right] = \theta^T \mu^\pi(s,a)$
- $\mu^\pi(s,a)$ feature expectation can be model as the discounted measure of features accorrding to system visitation frequency, given m trajectories (H^{i} is the length of the i^{th} trajectorie), $\mu^\pi(s,a)$ can be modeled as:

$$\mu^\pi(s,a) = \frac{1}{m} \sum_{i=0}^{m} \sum_{t=0}^{H_i} \gamma^i \phi\left(s_t^i, a_t^i\right)$$

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

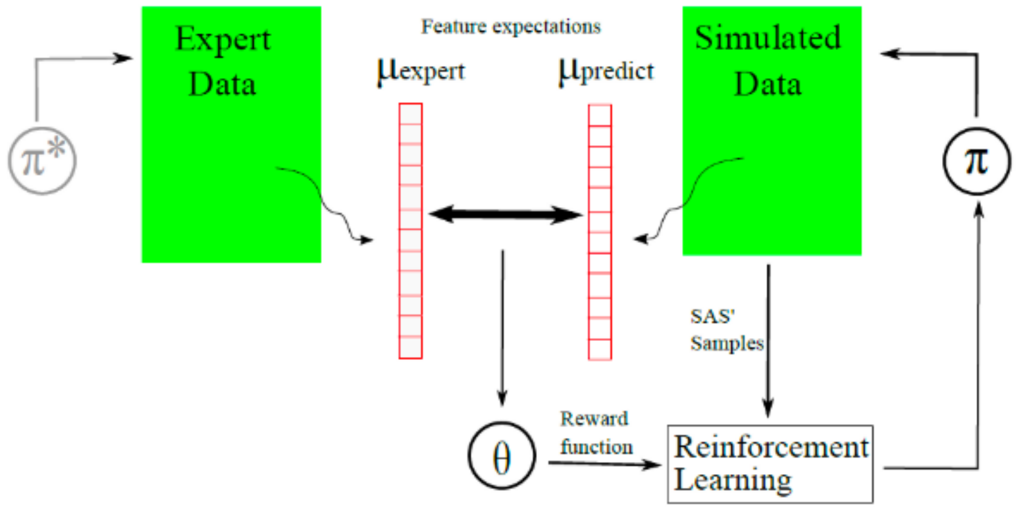
Algorithm

Algorithm 1 User simulation using imitation learning

1: Compute μ_{expert} from dialogue corpus
2: Initiate Π with random policy $\pi_{\text{predict}} = \pi_0$ and compute μ_{predict}
3: Compute t and θ such that

$$t = \max_{\theta}\{\min_{\pi_{\text{predict}}\in\Pi} \theta^T(\mu_{\text{expert}} - \mu_{\text{predict}})\} \text{ s.t. } \|\theta\|^2 \leq 1$$

4: **if** $t \leq \xi$ **then** Terminate
5: **end if**
6: Train a new policy π_{predict} for userMDP optimizing $R = \theta^T\phi(s,a)$ with RL (LSPI).
7: Compute μ_{predict} for π_{predict}; $\Pi \leftarrow \pi_{\text{predict}}$
Goto to step 3.



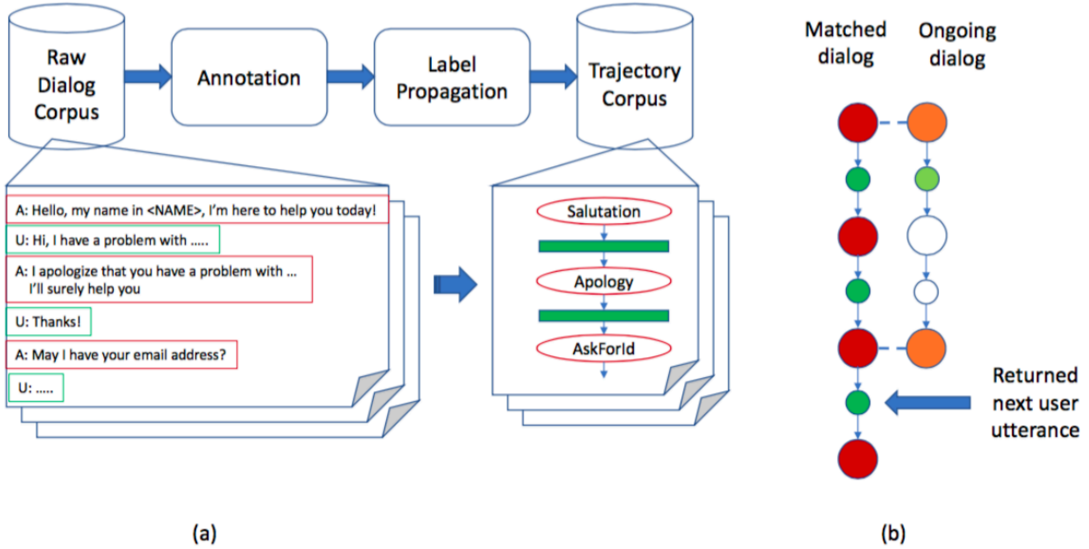
Summary

- We can train a MDP SU from a fix corpus
- In the paper, they only conducted a simple experiment
- The cost of computing is a lot. (RL in the inner-loop)

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Collaboration-based (Didericksen et al. 2017)

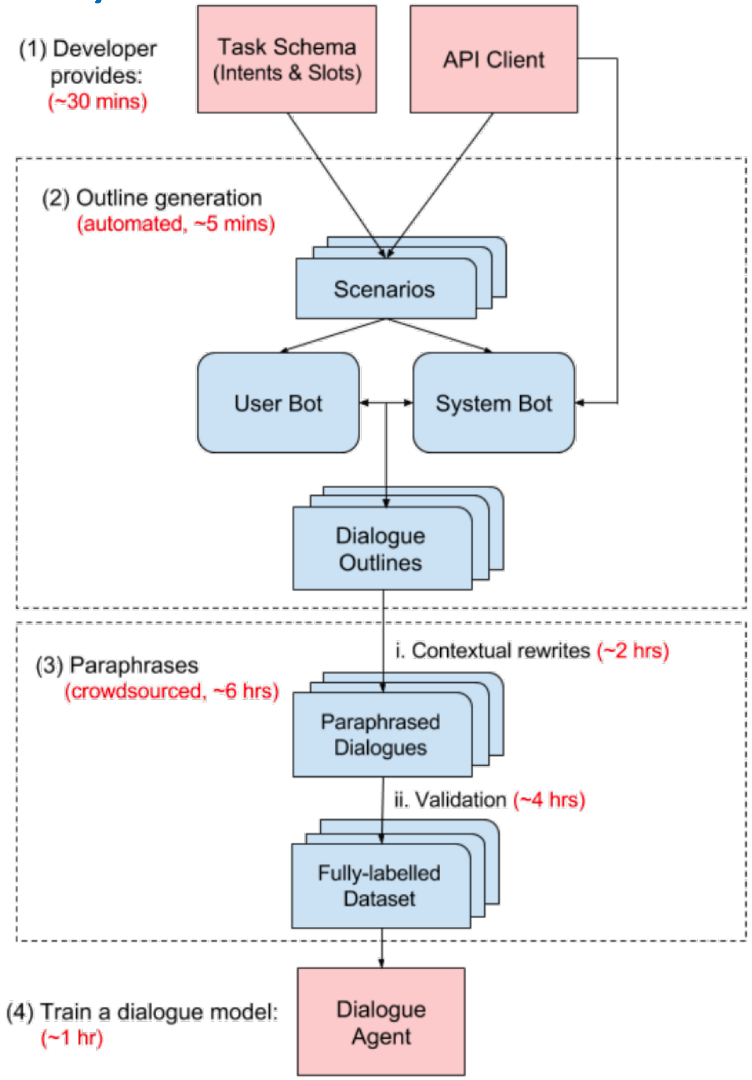
- Collaboration-based SU utilizes the similarity between different users to predict the user’s next action
- Label propagation:
  train a simple classification model on a part of the data to label the entire dataset
- Easy to incorporate external knowledge, e.g. user profile to pre-filter the act candidates
- Can be run very fast



HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

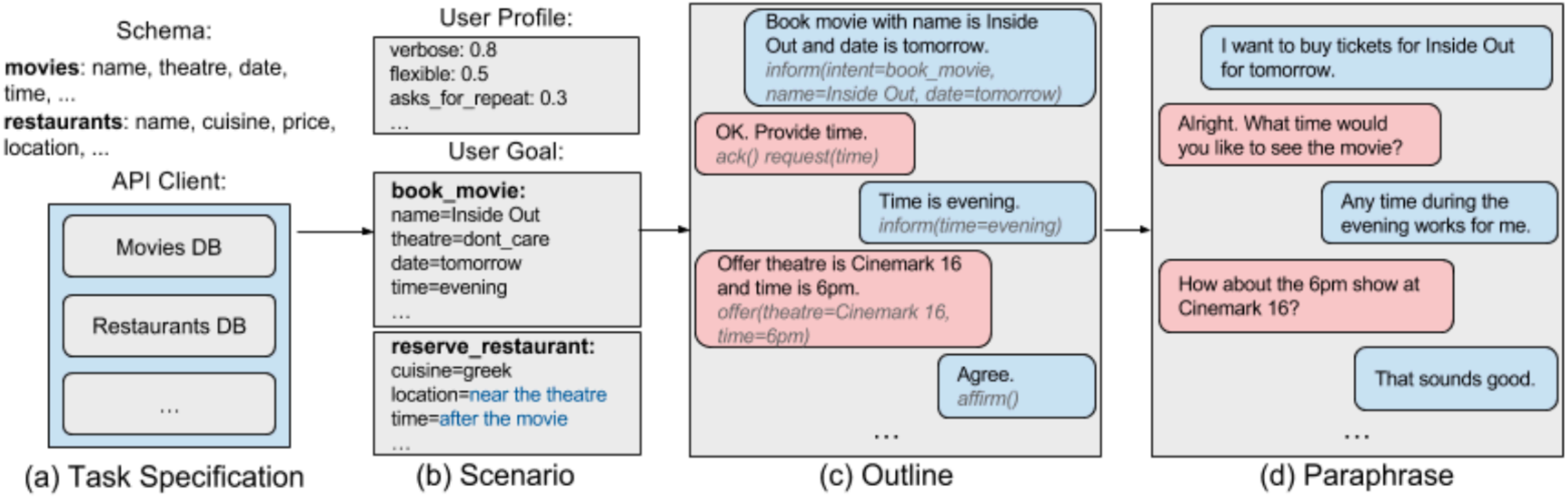
Build a Conversational Agent Overnight (Shah et al. 2018)

- Build a dialogue system by M2M and crowdsourcing
- Collect daya by Wizard-of-Oz setup may suffer from
  - Not cover all the interactions
  - Unfitting dialogues (too simplistic or too convoluted)
  - Need more efforts to filter errors



Generating outline via self-play

- Outlines are easier to generate
- Don't need to generate complex and diverse language



Conclusion

The rule-based methods

- ✓ More controllable
- ✓ Generate all possible paths
- – Domain-dependent
- – Not scalable
- – Labor-consuming

The model-based methods

- ✓ Learn user behaviour from corpus
- ✓ Less labor effort
- ✓ Adapt to new domain easilier
- – Focus on main paths, not all
- – Incoherence goal

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

What’s next?

- Generate more various outputs and more humain-like behaviour
- Persona for SU
- Error models: ASR, ambiguity, ...
- How to use IRL, adversarial training for SU?
- Self-training via Machine-to-machine interaction

Reference

- User modeling for spoken dialogue system evaluation
  Eckert, Wieland, Esther Levin, and Roberto Pieraccini, 1997
- HUMAN-COMPUTER DIALOGUE SIMULATION USING HIDDEN MARKOV MODELS
  Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon and Hiroshi Shimodaira. 2005
- Training Bayesian networks for realistic man-machine spoken dialogue simulation
  Olivier Pietquin, Stéphane Rossignol, and Michel Ianotto, 2009
- Probabilistic simulation of human-machine dialogues
  Scheffler, Konrad, and Steve Young, 2000
- Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System
  Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye and Steve Young, 2007
- A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems
  Layla El Asri, Jing He, Kaheer Suleman, 2016

Reference

- A User Simulator for Task-Completion Dialogues
  Xiujun Li, Zachary C. Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, Yun-Nung Chen, 2017
- Neural User Simulation for Corpus-based Policy Optimisation for Spoken Dialogue Systems
  Kreyssig F, Casanueva I, Budzianowski P, Gašić M, 2018
- USER MODELING FOR TASK ORIENTED DIALOGUES
  Izzeddin Gur, Dilek Hakkani-Tur, Gokhan Tur, Pararth Shah, 2018
- How to Build User Simulators to Train RL-based Dialog Systems
  Weiyan Shi, Kun Qian, Xuewei Wang, Zhou Yu, 2019
- User Simulation in Dialogue Systems using Inverse Reinforcement Learning
  Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, Olivier Pietquin, 2011
- Collaboration-based User Simulation for Goal-oriented Dialog Systems
  Devin Didericksen, Oleg Rokhlenko, Kevin Small, Li Zhou, Jared Kramer, 2017

Reference

- Building a Conversational Agent Overnight with Dialogue Self-Play
  Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, Larry Heck, 2018