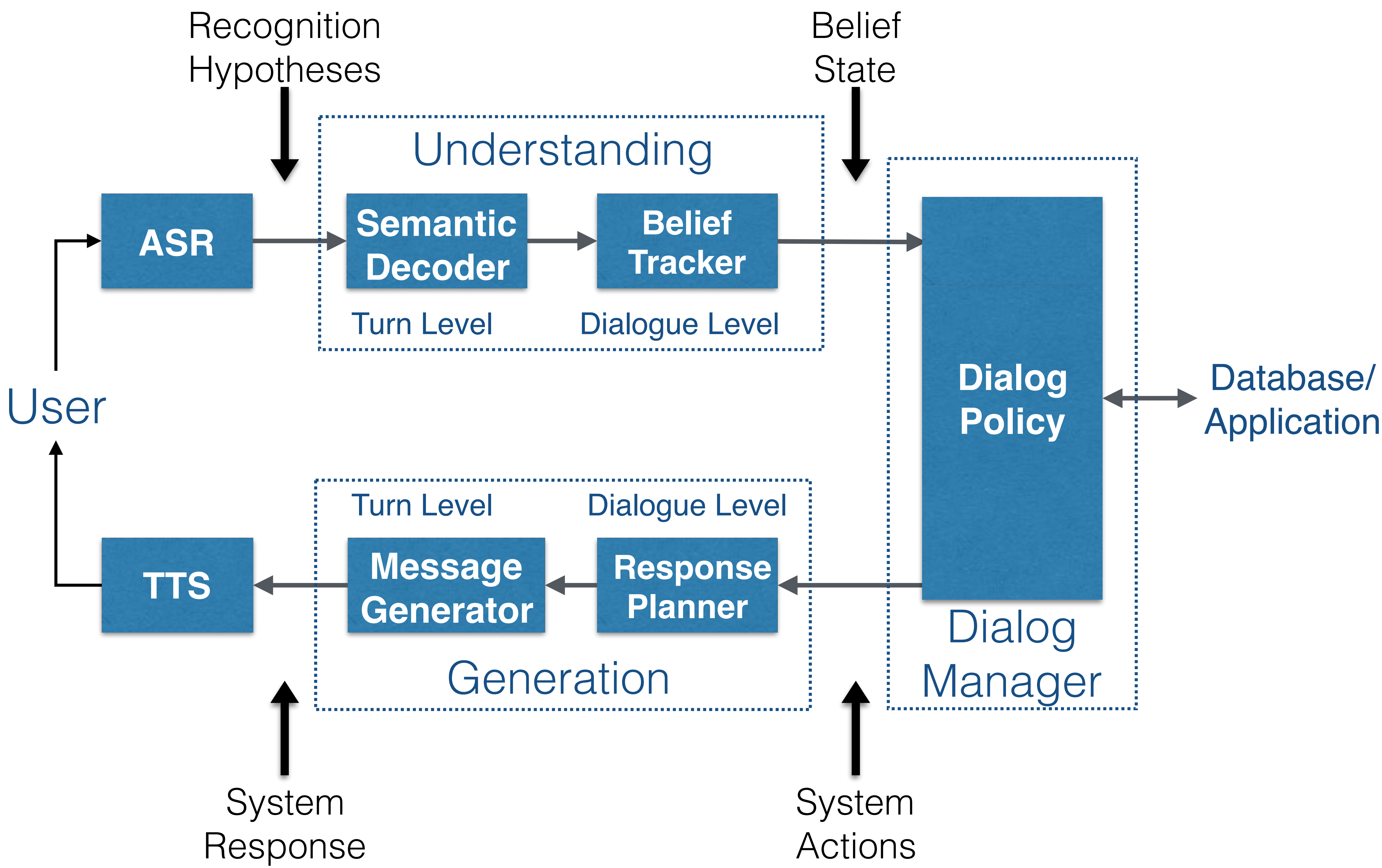
Applications of Deep Learning in Spoken Dialogue Systems

Steve Young

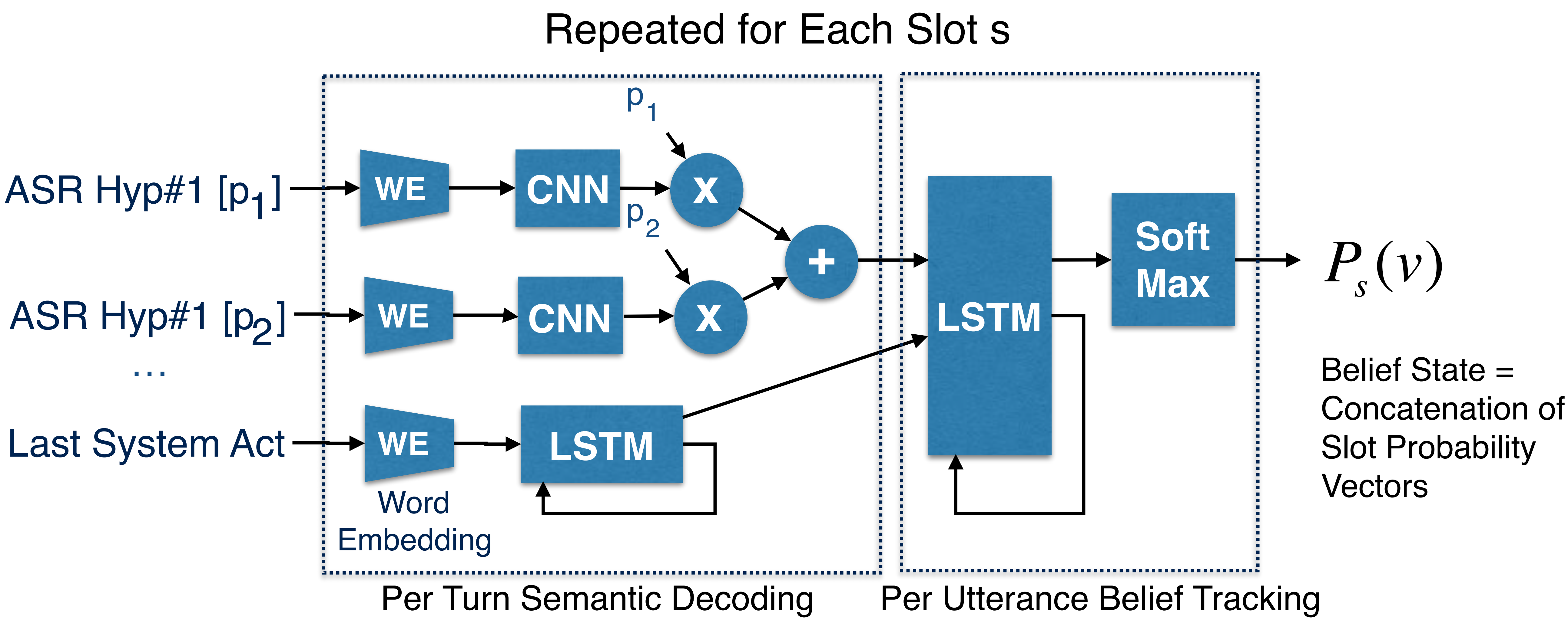
Engineering Department
Trumpington Street
Cambridge, UK
sjy@eng.cam.ac.uk

Apple Europe Limited
90 Hills Road
Cambridge, UK
steve_young@apple.com

Dialog System Architecture



Understanding: ASR -> Beliefs



Henderson, M., et al. (2014). Word-Based Dialog State Tracking with Recurrent Neural Networks. SigDial 2014, Philadelphia, PA.

Rojas-Barahona, L., et al. (2016). Exploiting Sentence and Context Representations in Deep Neural Models for Spoken Language Understanding. Coling, Osaka, Japan.

Mrksic, N., et al. (2016) Neural Belief Tracker: Data-Driven Dialogue State Tracking. arXiv:1606.03777

Generation: actions -> words

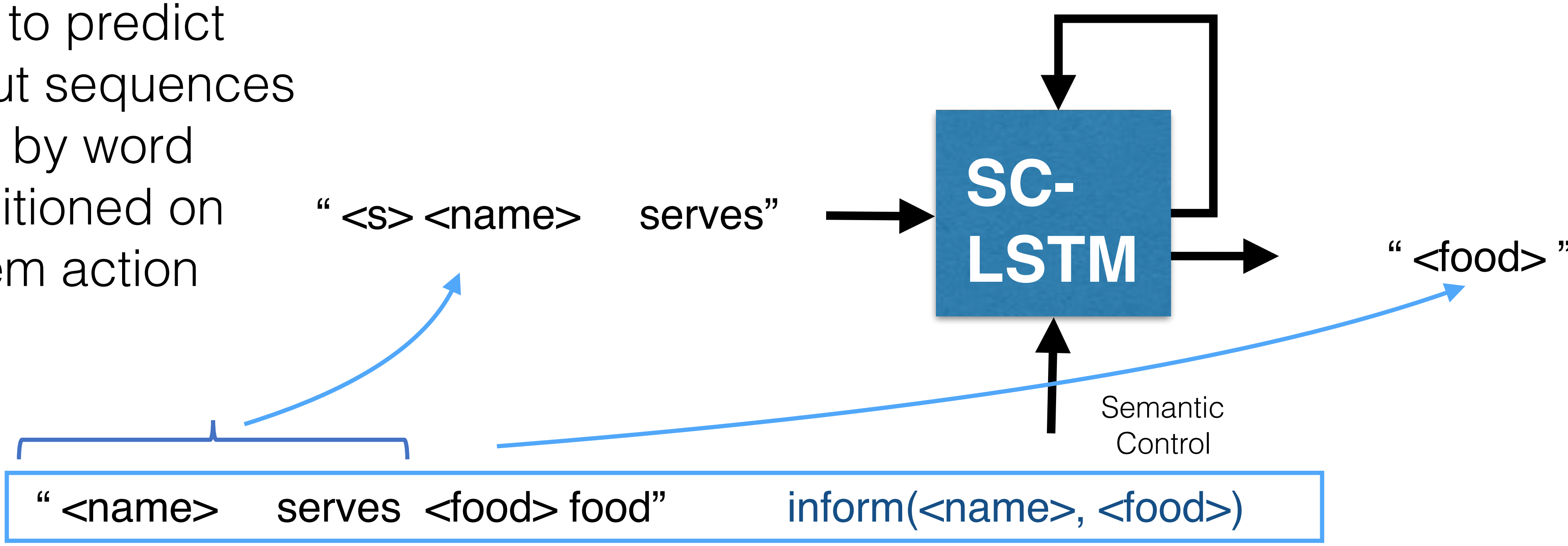
Need to convert abstract system actions to natural language e.g.

inform(name=“The Peking”, food=“chinese”) → “The Peking serves chinese food”

Solution: delexicalise the training data, and train a conditional LSTM

inform(name=<name>, food=<food>) → “<name> serves <food> food”

Train to predict output sequences word by word conditioned on system action



Generation: actions -> words

Need to convert abstract system actions to natural language e.g.

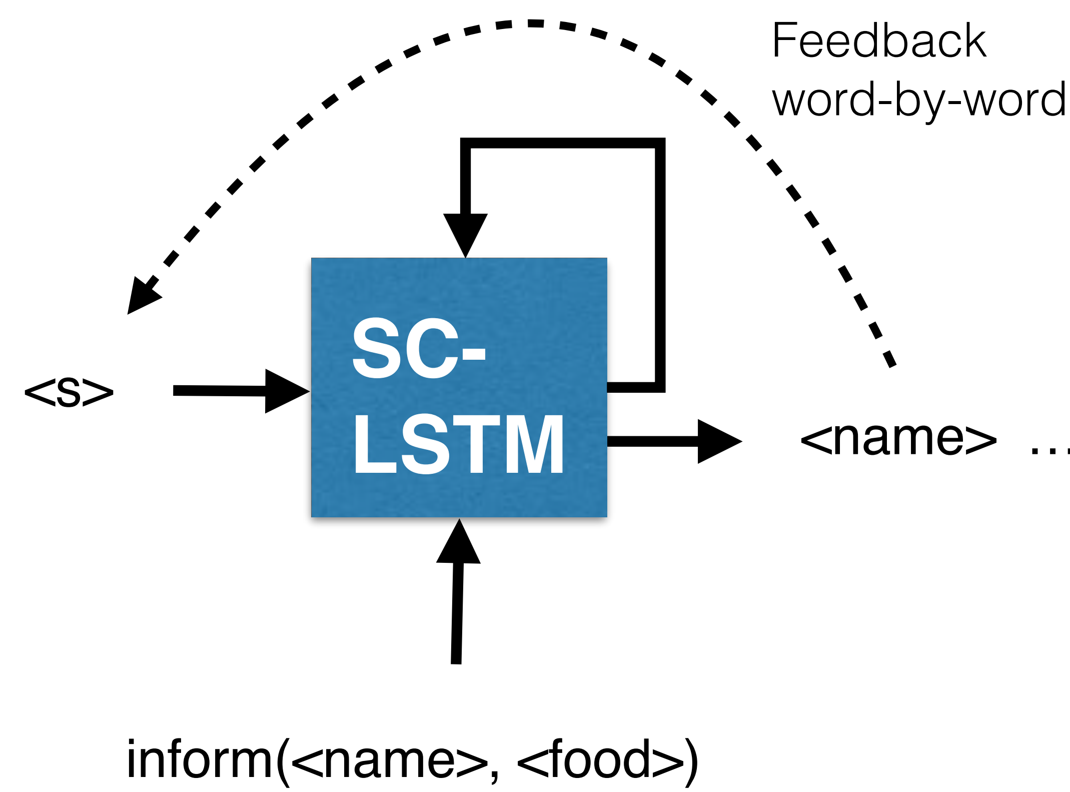
inform(name="The Peking", food="chinese") → "The Peking serves chinese food"

Solution: delexicalise the training data, and train a conditional LSTM

inform(name=<name>, food=<food>) → "<name> serves <food> food"

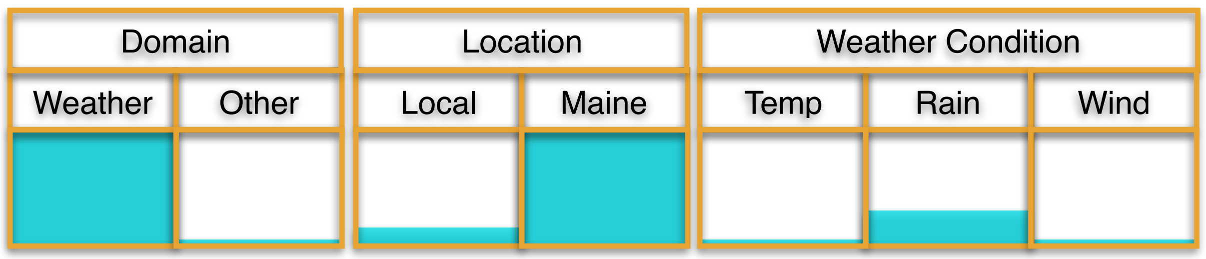
At runtime, condition with system action and prime with start symbol …

… then re-lexicalise.



T-H. Wen et al (2015). "Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems." EMNLP 2015, Lisbon, Portugal.

Dialog Manager



$$\mathbf{b} \xrightarrow{\pi} a$$

Actions: request, confirm, inform, execute, etc

1. Belief state $\mathbf{b}$ encodes the state of the dialog, including all relevant history.
2. Belief state is updated every turn of the dialog.
3. The policy π determines the best action to make at each turn via a mapping from the belief state $\mathbf{b}$ to actions a.
4. Every dialog ends with a reward: +ve for success, -ve for failure. Plus a weak -ve reward for every turn to encourage brevity.
5. Reinforcement Learning is used to find the best policy.

Reinforcement Learning

Policy: $\pi(\mathbf{b},a):\mathbb{R}^n \times A \rightarrow [0,1]$

Reward: $R=\sum_{\tau=1}^{T} r(\mathbf{b}_\tau,a_\tau)$

Value: $Q_\pi(\mathbf{b}_t,a_t)=\mathbb{E}_\pi\left[\sum_{\tau=t+1}^{T} r(\mathbf{b}_\tau,a_\tau)\right]$

Problem:

$\boxed{\text{Find optimal policy} \quad \pi^* = \underset{\pi}{\arg\max}\ \{E[R\mid\pi]\}}$

or

$\boxed{\text{Solve} \quad Q^*_\pi(\mathbf{b}_t,a_t) = r_{t+1} + \max_a \{Q^*_\pi(\mathbf{b}_{t+1},a)\}}$

Implementation Algorithms

Policy Gradient

Advantage Actor Critic (A2C)

Trust Region Actor Critic (TRACER)

Natural Actor Critic (eNACER)

Value Iteration

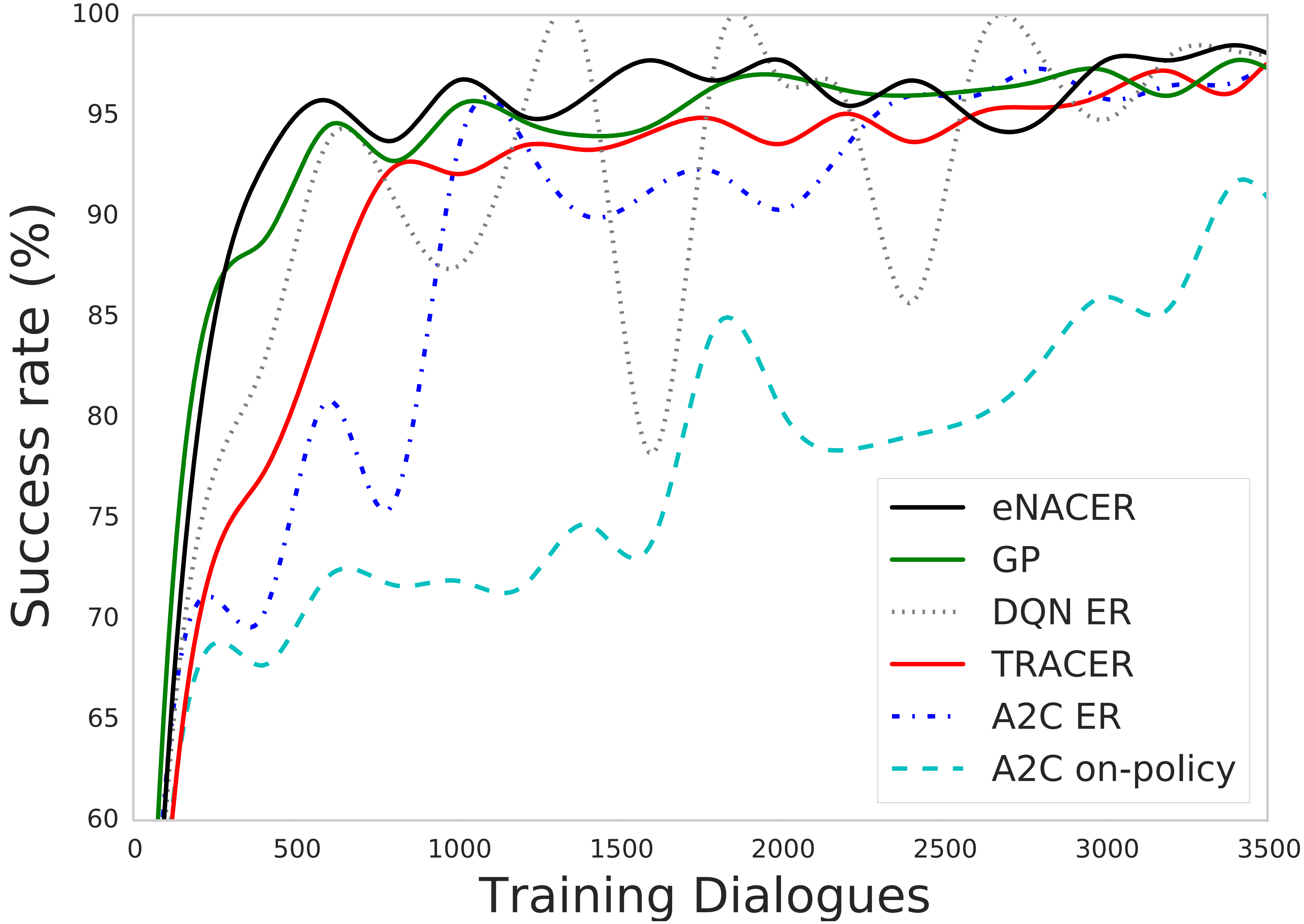
Deep Q Network (DQN)

[Gaussian Process (GP)]

Y. Li (2017). "Deep Reinforcement Learning: An Overview." arXiv:1701.07274v2.

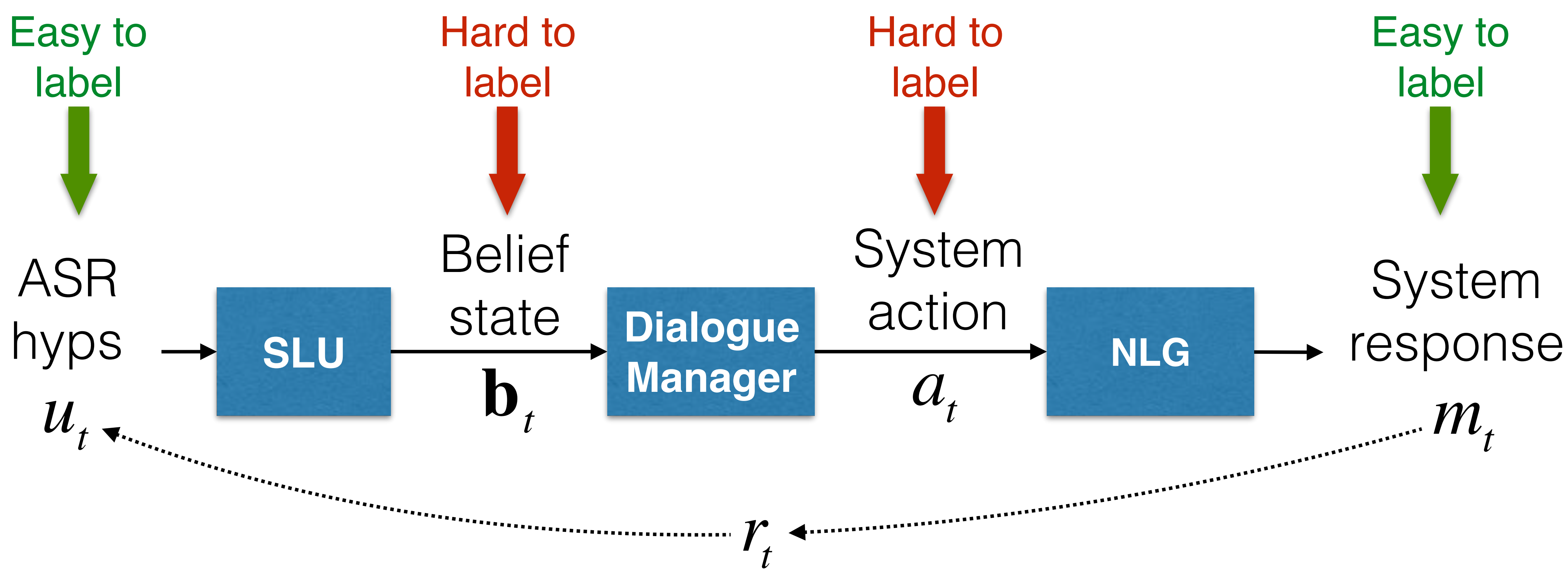
See also David Silver's 2016 ICML Tutorial.

Typical Early Learning



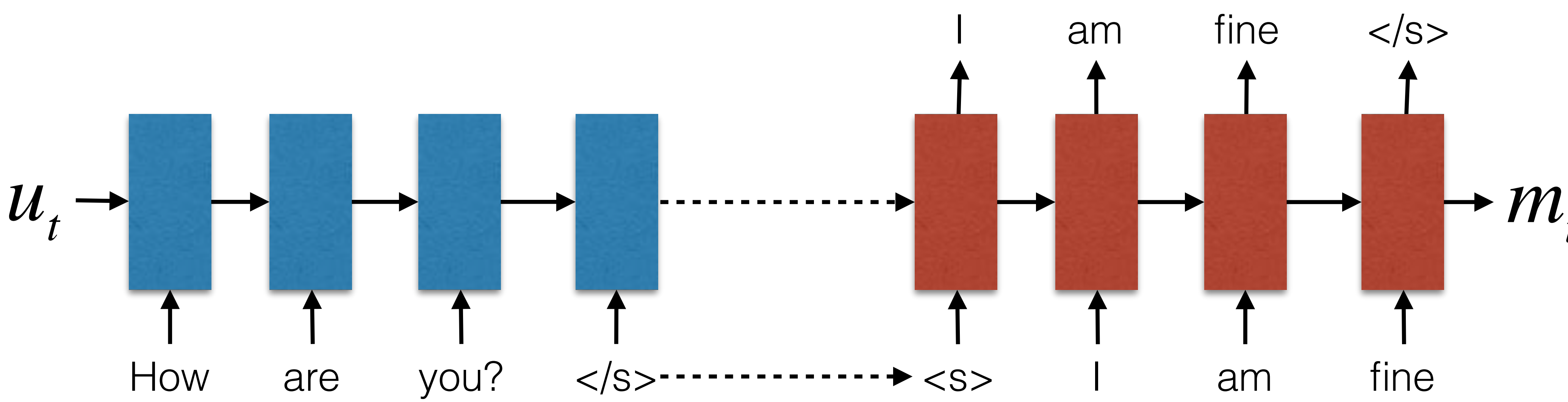
P.-H. Su, P. Budzianowski, S. Ultes, M. Gasic and S. Young (2017). "Sample-efficient Actor-Critic Reinforcement Learning with Supervised Data for Dialogue Management." SigDial, Saarbrucken, Germany.

The Labelling Problem



So can we train End-To-End using just $\langle u_t, m_t, r_t \rangle$?

Sequence to Sequence models



LSTM Encoder

LSTM Decoder

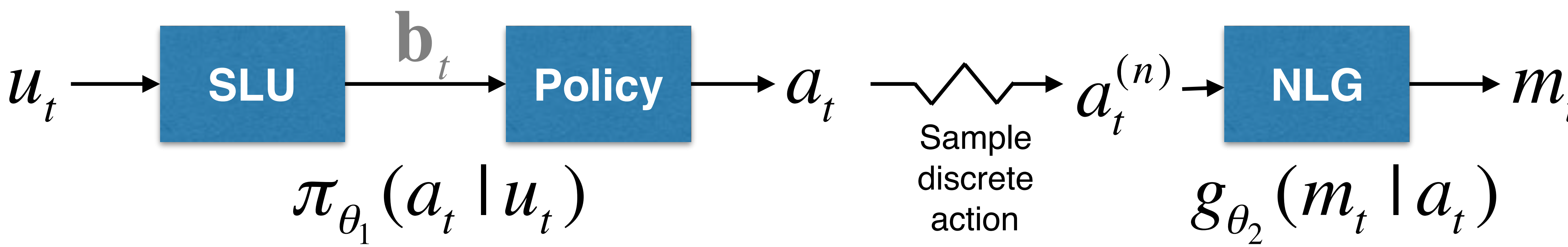
Supervised Learning (no reward): maximise logP of correct response m given input u for every input/output pair in training set.

$$L(\theta) = \sum_{<u_i, m_i>} \log P(m_i \mid u_i)$$

Good for chatbots, but no explicit knowledge base and no planning

Sutskever, O. Vinyals and V. Le (2014). "Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." NIPS.
O. Vinyals and Q. Le (2015). "A Neural Conversational Model." ICML Deep Learning Workshop.

Multicomponent System End-To-End Training

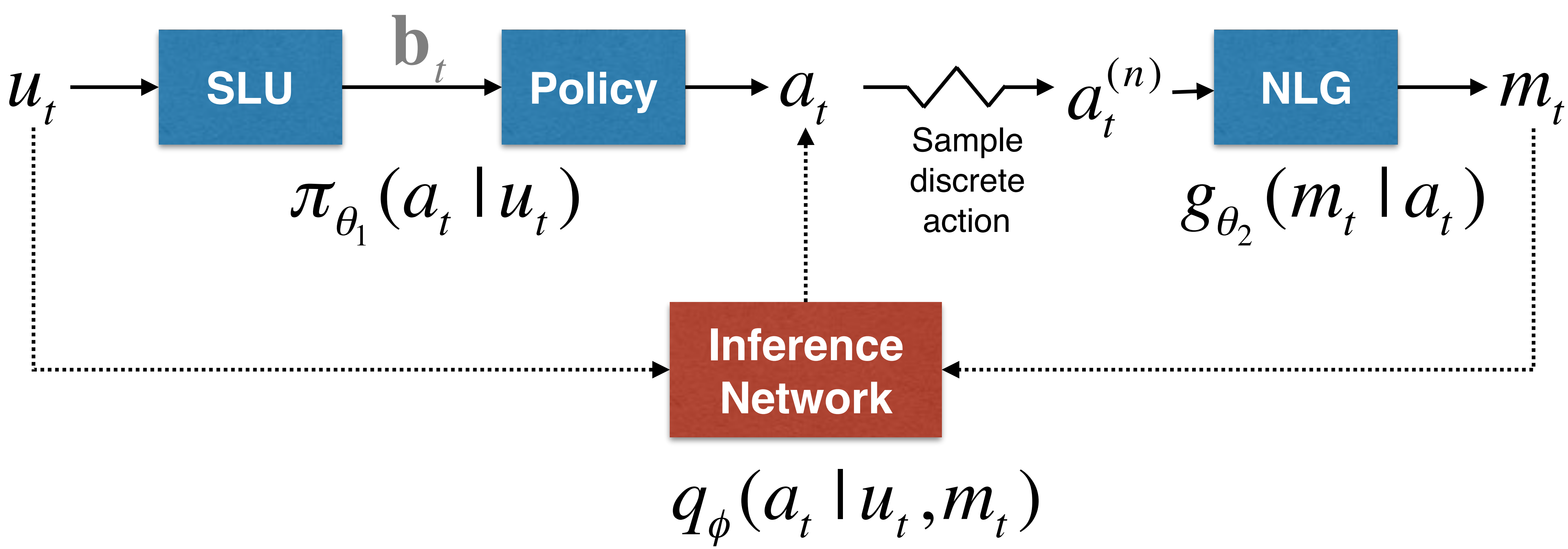


The action a_t is now a discrete latent variable

$$p(m_t \mid u_t) = \sum_a g(m_t \mid a)\pi(a \mid u_t)$$

Unfortunately, there is no tractable way to compute this inference and Monte Carlo methods are too slow.

Neural Variational Inference (NVI)



Maximise the variational lower bound

$$L(m,u,\theta_1,\theta_2,\phi) = \mathbb{E}_q\left[\log g(m_t \mid a_t)\right] - \lambda D_{KL}\left(q(a_t) \mid \pi(a_t \mid .)\right)$$

Mnih and K. Gregor (2014). "Neural Variational Inference and Learning in Belief Networks." ICML, Beijing, China.

NVI Optimisation

1) Randomly sample a minibatch of training data

$$\mathbb{D} = \langle u_1, m_1 \rangle \ldots \langle u_N, m_N \rangle$$

2) For each $\langle u_i, m_i \rangle$, generate N samples from inference net

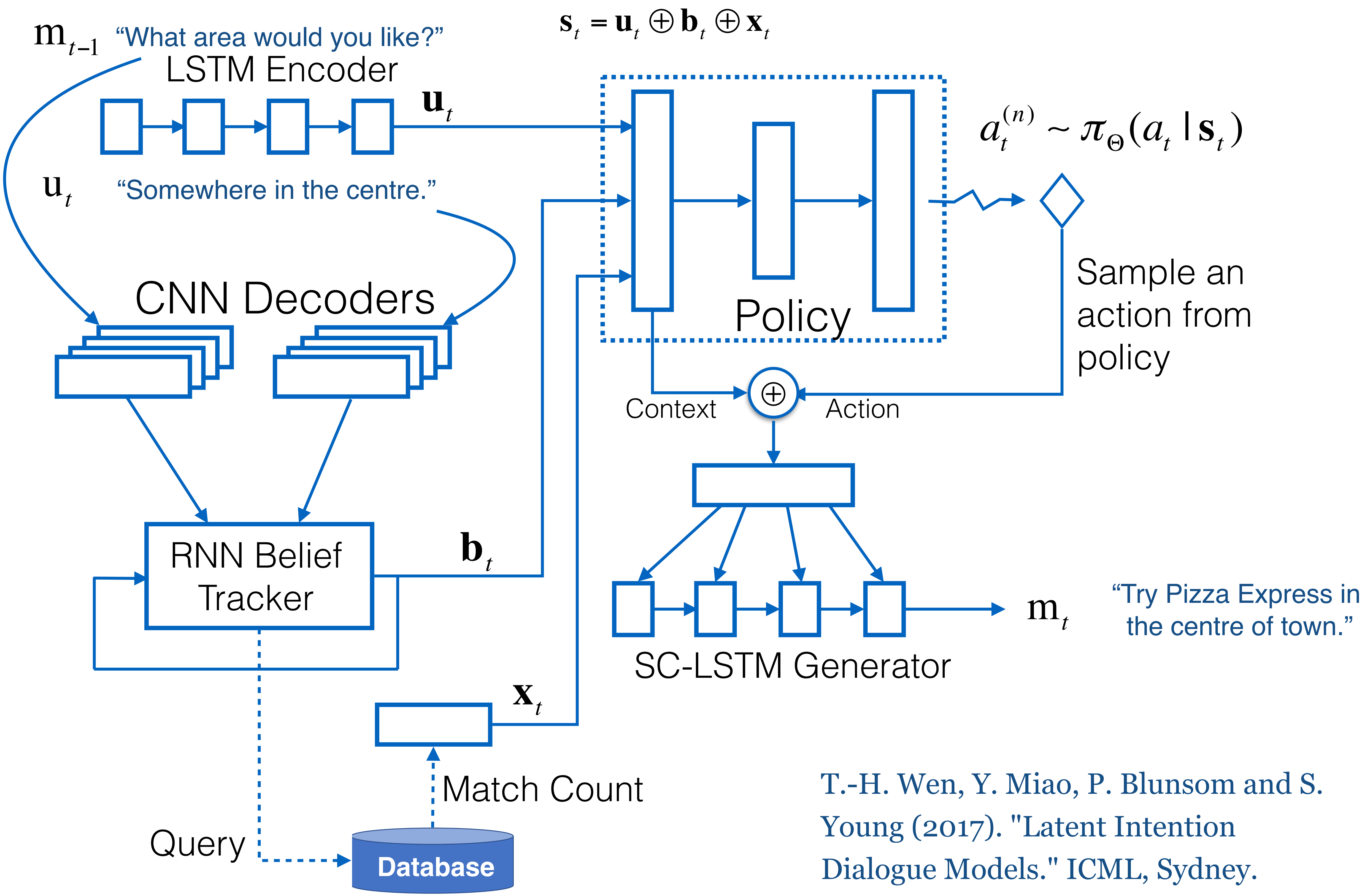
$$a_i^{(1)} .. a_i^{(N)} \sim q_\phi(a \mid u_i, m_i)$$

3) Compute gradients $\nabla_\theta L$ and $\nabla_\phi L$ using Monte Carlo integration to estimate expectations.

4) Update parameters

Note that $\nabla_\phi L$ is very noisy and variance reduction techniques are required in practice

Latent Intention Dialogue Model



T.-H. Wen, Y. Miao, P. Blunsom and S. Young (2017). "Latent Intention Dialogue Models." ICML, Sydney.

Training the LIDM

In practice, learning latent actions in a completely unsupervised manner is extremely difficult. Hence, a multi-stage approach was taken for training the full end-to-end dialogue system:

- the SLU Component was pre-trained on labelled data
- Part of the training corpus was clustered to provide a subset of automatically labelled actions
- Variational lower bound maximisation was interleaved with supervised learning on the automatically labelled data
- Reinforcement learning was used to fine tune policy parameters

T.-H. Wen, Y. Miao, P. Blunsom and S. Young (2017). "Latent Intention Dialogue Models." ICML, Sydney.

Sample Dialogue

Human :	Hi, I'm hungry for some Indonesian		
Machine:	**(0**	**0.71)**	**i am sorry , there are no indonesian restaurants in the area . would you like something else ?**
	(46	0.19)	there are no restaurants serving indonesian kind of food .
	(34	0.07)	what location would you like ?
Human :	Oh no. How about Indian?		
Machine:	(0	0.59)	there are 9 restaurants in the east side of town . would you like their address ?
	(30	0.15)	curry prince serves indian cuisine .
	(34	**0.10)**	**what area would you like ?**
Human :	East side please		
Machine:	**(0**	**0.62)**	**curry prince is a indian restaurant in the east side of town . would you like their telephone number ?**
	(84	0.19)	curry prince is an indian restaurant in the east of town and the number is 01223 566388 .
	(81	0.08)	curry prince serves indian type of cuisine and is located in the east part of town .
Human :	yes , the address and phone number please		
Machine:	(3	0.46)	the address is 451 newmarket road fen ditton and the phone is 01223 566388 .
	(11	**0.16)**	**the address for curry prince is 451 newmarket road fen ditton and the phone number is 01223 566388 .**
	(0	0.15)	curry prince is located at 451 newmarket road fen ditton and their phone is 01223 566388 . is there anything else i can help you with ?
Human :	Okay, thank you goodbye		
Machine:	(0	0.57)	thank you , goodbye .
	(2	0.13)	goodbye .
	(1	**0.11)**	**you are welcome . goodbye .**
	(6	0.10)	thank you for using the cambridge restaurant system . goodbye .

actual outputs selected in dialogue shown in bold

Summary

- DNNs provide a flexible building block for all stages of the dialogue system pipeline, though training is rarely as straightforward as research papers suggest!
- Labelled data is expensive and each stage of a multi-component pipeline requires its own labelled data set.
- "End-2-End" multi-component training has potential to reduce labelled data requirement and potentially avoid hand-crafting internal interfaces.
- Users can provide feedback for free but the feedback signal is weak and noisy. Reinforcement Learning provides a framework for exploiting this mostly untapped resource.

Credits

All members of the Cambridge Dialogue Systems Group
Past and Present:

Milica Gasic	Matt Stuttle
Catherine Breslin	Martin Szummer
Pawel Budzianowski	Eddy Su
Matt Henderson	Blaise Thomson
Filip Jurcicek	Pirros Tsiakoulis
Simon Keizer	Stefan Ultes
Dongho Kim	David Vandyke
Fabrice Lefevre	Karl Weilhammer
Francois Mairesse	Shawn Wen
Nikola Mrksic	Jason Williams
Lina Rojas Barahona	Hui Ye
Jost Schatzmann	Kai Yu